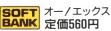
1990年12月1日発行（毎月1回1日発行）第9巻12号通巻104号　昭和58年11月2日第三種郵便物認可

特集 XCのための傾向と対策

THE SOFTOUCH SPECIAL II 特大GAME REVIEW
シュミレーションプログラミング入門
特別企画アナログジョイスティックの製作

12
1990

オー！エックス
定価560円

SHARP

X68000を象徴する
高解像度自然色グラフィック

X68000の数々の注目すべき能力のうち、とりわけ際立っているのが「強力なグラフィック能力」だといえましょう。流通する多くのコンピュータが8色表示という環境の中で、65,536色同時表示(512×512ドットモード時)というハイスペックで登場。発売後、数年経過した現在でも初代X68000、CZ-600Cカタログの表現が、そのまま通用します。「**クロームやチタニウムに代表される高品位な金属の質感、金、銀表現、人の目に映る色や形状をほとんどありのままに表現し得る自然色グラフィックスが、これまでのCGイメージを一新します。**」……まさにX68000を象徴するこのスペックは、ごく最近までパーソナルコンピュータでは不可能とされた"レイトレーシング"にしても、フレームバッファなどの増設なしで実現可能とし、その設計思想の先見性と、コストパフォーマンスの高さを雄弁に物語っています。

65,536色C.G.▲

もう一歩ふみこんだ
オリジナルアートの世界を

65,536色を駆使するには、カラーイメージユニット(CZ-6VT1)やカラーイメージスキャナ(CZ-8NS1)などの画像入力機器が役立ちます。これらの機器は写真、水彩画、油絵、色鉛筆やパステルなど、ビジュアル化のあらゆるアプローチをディスプレイ上にシミュレートし、それぞれの特徴を活かしながら、独自のCGの世界を創造します。まさにAVに強いと言われるX68000の真骨頂を示すものといえましょう。また一般に色数が少ないとされるアニメ調の絵にしても、肌色などの中間色に対する妥協を許しません。

各種ペイントの微妙なタッチも再現 ▲

▲65,536色モードと
◀8色の比較

クリエイティブワークを広げる
先駆の独立3画面設計

そしてX68000のメモリアーキテクチャは、68000CPUのもつ広大なアドレス空間を活かして、テキスト、グラフィック、スプライトの3画面を独立構造として装備した独自の画面設計です。文字、CG、キャラクタをプライオリティつきで重ね合わせ表示する、これまでむずかしかったビジュアル表現も造作なくこなすハイアビリティがクリエイティブワークの幅を広げます。なかでもゲームデザインを強力にサポート、アニメーションと呼ぶにふさわしい興奮のシーンを展開できるスプライトとバックグラウンドはクリエイターの血を騒がせています。またパソコンテレビX1の血統を受け継いだスーパーインポーズ機能、テレビやビデオの映像をバックに通信したりプログラミングにチャレンジしたり……こうした環境を標準で楽しめるところにも、X68000の

▲65,536色画像取り込みを迫力の21型ディスプレイで…。

敢えてX68000のグラフィックアビリティを実証する意味。

能力の一端をかいまみることができます。

▲映像をバックに通信も

フレキシブルなビットマップ方式のテキストRAM

もちろん512KバイトVRAMを搭載したテキスト機能も、その特色の一つに数えられます。一般に使用される80桁表示をこえた96桁のテキスト表示(ANK)もさることながら、ビットマップ方式の採用により、各キャラクタのみでなく1ドット毎の細かい制御が可能(MELT.R)。加えてユーザー独自の字体をデザインして使用することも比較的容易です。また、このテキストVRAMはグラフィックの表示も可能であり、SX-WINDOWのPIX形式のファイルやPDSのMAKI形式で利用されており、他機種の8/16色グラフィックとのデータの共有を実現しています。このテキストVRAMを利用してオリジナルのTITLE.SYSをつくったりスーパーインポーズ機能を活かしてTVパソコンとして楽しんだり、カスタマイズの世界がどんどん広がっていきます。2、3のアプリケーションで納得するパソコン、それはそれとして、ここには文房具としてでなく創造するものとしてのパソコンの世界があります。

68000CPUとX68000の可能性は、まだほんの頭を覗かせただけかもしれません。

テキストのドットコントロール(MELT.R)▲

ユーザーフォント作成例▲

オリジナルTITLE.SYS▶

レイトレーシング:立体感のあるC.G.を描画するため、目から光源へさかのぼって反射鏡をシミュレートする技法。光線追跡法ともいう。かつてはその計算に数日を要したが、今では高速演算プロセッサとの組み合わせでスピーディに処理でき、パソコンレベルでもチャレンジ可能。

フレームバッファ:画像データを記憶するメモリ(グラフィックRAM)、フレームメモリともいう。通常、パソコンの拡張スロットに差し込んで用い、高度なCGを実現できるが、高価なのでちょっと手が出しづらい。

中間色:肌色や金属色など、8色や16色表示では表現しにくい色調。

広大なアドレス空間:多くの機種では、CPU(中央制御装置)が巨大なメモリを扱えてもOS(オペレーティングシステム)の関係で使えるメモリが限られるという矛盾した状態になる。もちろんX68000ではそのようなことはない。

テキスト:文字や記号だけを表示する画面。通常は16×16ドットのキャラクタを1文字単位でコントロールする。

グラフィック:絵を書くための画面。1ドット毎にコントロールでき、ドット(点)の集合として描画する。

スプライト:グラフィック画面上で重ね合わせをする機能。ゲームのキャラクタなどを高速で動かすための機能のひとつ。X68000では、水平32スプライト、1画面128スプライト同時表示というハイスペックを実現している。

VRAM:ブイラムあるいはビデオラム。CRTディスプレイに表示するためのメモリ。RAMの内容に応じて、文字やグラフィックが表示されるように処理されたメモリ回路。

ANK:Alphabet,Numeric,Kana。アンクと読む。アルファベット、数字、かなの3種類の半角文字をANK文字と呼んでいる。

MELT.R:On氏作のPDS。オリジナルアイデアは他機種のもの。参考までに、このプログラムでは、文字の各ドットは操作しておらず、表示位置をドット単位で操作する。

ユーザー独自の字体:丸文字、変体少女文字に始まって、最近ではロシア語に至るまで、様々な字体が広がっている。従来の「外字」は、個人のものであったが、通信の広がりにより同じフォントを使うといった事態も生じている。

MAKI形式:草の根ネット「まきちゃんNET」において開発された640×400ドットアナログ16色画像のデータファイル用フォーマット。X1/turbo、MZ25/28など多くの機種間での画像ファイル共用の架け橋となる。

TITLE.SYS:通常は、電源を入れると「SAKO-DOS V2.1 Copyright1990 SAKORIN」などといった渇いたメッセージが表示されるが、X68000の場合は、画像がこれのかわりを勤める。これが「TITLE.SYS」であり、知識があれば自分だけのTITLE.SYSも作れる。

【今回の広告に使用したツール一覧】SCANTOOL.X(シャープ製)、MONO.X/GtoT8.X/TITLE.X/GTCONV.X/TXCLR.R(WOODY RINN氏作)、MFGED.X/MFLOAD.X(結城 見氏作)、GL3LD.X/(HoNDA氏作)、ACF.X(夢職人氏作)、MuTerm.X(はちくん氏作)、PIC.R(柳沢 明氏作)、MELT.R(On氏作)

ゴメンナサイ!! ●ADPCMの英文中、「Code」の文字が欠落しておりました。また●「アドレスバスは16ビットながら……」などと大変な間違いをしでかしてしまい、広島県の山本様ほか多数の皆様からご指摘を受け、日々反省しております。正しくは「データバス……」でした。

SUPER HD
本体+キーボード+マウス・トラックボール
CZ-623C-TN(チタンブラック) 標準価格498,000円(税別)

EXPERT II
本体+キーボード+マウス・トラックボール
CZ-603C-BK(ブラック)・-GY(グレー) 標準価格338,000円(税別)
HDタイプ CZ-613C-BK(ブラック) 標準価格448,000円(税別)

PRO II
本体+キーボード+マウス
CZ-653C-BK(ブラック)・-GY(グレー) 標準価格285,000円(税別)
HDタイプ CZ-663C-BK(ブラック)・-GY(グレー) 標準価格395,000円(税別)

充実のディスプレイ ラインアップ
※印の商品は在庫僅少です。

15型カラーディスプレイテレビ(ドットピッチ0.39㎜) CZ-602D-BK(ブラック)・-GY(グレー)※
標準価格 99,800円(チルトスタンド同梱・税別)
15型カラーディスプレイテレビ(ドットピッチ0.39㎜) CZ-605D-BK(ブラック)・-GY(グレー)
標準価格115,000円(スピーカー2個/チルトスタンド同梱・税別)
15型カラーディスプレイテレビ(ドットピッチ0.31㎜) CZ-613D-TN(チタンブラック)・-BK(ブラック)・-GY(グレー)
標準価格135,000円(スピーカー2個/チルトスタンド同梱・税別)
14型カラーディスプレイ(ドットピッチ0.31㎜) CZ-603D-BK(ブラック)・-GY(グレー)
標準価格 84,800円(チルトスタンド同梱・税別)
14型カラーディスプレイ(ドットピッチ0.31㎜) CZ-604D-BK(ブラック)・-GY(グレー)
標準価格 94,800円(スピーカー2個/チルトスタンド同梱・税別)
21型カラーディスプレイ(ドットピッチ0.52㎜) CU-21HD-BK(ブラック)
標準価格148,000円(スピーカー2個同梱・税別)

X68000 自分流カード デザインコンペ 作品大募集

〈応募要領〉●応募方法/X68000で作成したポストカードサイズのデザインカードを送って下さい。(ソフトは自由)●作品分類/部門A:クリスマスカード、ニューイヤーカード 部門B:バレンタインカード、バースデイカード 部門C:暑中見舞カード、サークル・趣味の会お知らせカード●賞/A・B・C各部門毎に優秀作品を選考、オリジナルカレンダーに掲載してプレゼントします。※優秀作品賞:掲載作品応募者に、カレンダー及びオリジナル表彰楯を進呈。※参加賞:応募者全員に、カレンダーを進呈。(応募作品に関わる諸権利は主催者に帰属するものとして作品の返却はいたしません)
●応募期間/1990年10月1日〜1991年2月28日(消印有効)

詳細はX68000販売店店頭で、チラシ・ポスターをご覧下さい。

本広告に関するご意見をお寄せください。下記大阪本社宣伝部「☆さこう★係」まで

●お問い合わせは…
シャープ株式会社
シャープ㈱電子機器事業本部システム機器営業部
〒545 大阪市阿倍野区長池町22番22号☎(06)621-1221(大代表)
電子機器事業本部液晶映像システム事業部第2商品企画部
〒162 東京都新宿区市谷八幡町8番地☎(03)260-1161(大代表)

特集 XCのための傾向と対策

XBAStoC CHECKER

(で)のショートプロぱーてぃ

イメージファイト

NAIOUS

エアー・コンバット（遊撃王II）

Oh!X

C O N T

●特集

〈スタッフ〉
●編集長／前田 徹 ●副編集長／植木章夫 ●編集／岡崎栄子 浅井研二 ●協力／有田隆也 中森 章 後藤貴行 林 一樹 荻窪 圭 岡本浩一郎 毛内俊行 吉田賢司 影山裕昭 相馬英智 古村 聡 村田敏幸 丹 明彦 三沢和彦 長沢淳博 宮島 靖 金子俊一 浦川博之 山田純二 ●カメラ／杉山和美 ●イラスト／永沢しげる 山田晴久 小栗由香 ●アートディレクター／島村勝頼 ●レイアウト／元木昌子 AD GREEN ●校正／グループごじら

表紙絵：須藤　牧人

E N T S

1990 DEC. 12

■広告目次

SHARP

システムパフォーマンスを実証する多彩なペリフェラル。

ディスプレイ関連

カラーディスプレイテレビ

15型カラーディスプレイテレビ
CZ-602D-BK
★CZ-602D-GY
標準価格 99,800円(税別)
(チルトスタンド同梱)

カラーディスプレイ

14型カラーディスプレイ
CZ-603D-BK・-GY
標準価格 84,800円(税別)
(チルトスタンド同梱)

15型カラーディスプレイテレビ
CZ-605D-BK・-GY
標準価格 115,000円(税別)
(スピーカー2個・チルトスタンド同梱)

14型カラーディスプレイ
CZ-604D-BK・-GY
標準価格 94,800円(税別)
(スピーカー2個・チルトスタンド同梱)

15型カラーディスプレイテレビ
CZ-613D-TN・-BK・-GY
標準価格 135,000円(税別)
(スピーカー2個・チルトスタンド同梱)

21型カラーディスプレイ
CU-21HD
標準価格 148,000円(税別)
(スピーカー2個同梱)

CRTフィルター

高性能CRTフィルター
BF-68PRO
標準価格 19,800円(税別)
(14/15型用)

チューナー

RGBシステムチューナー
CZ-6TU-BK・-GY
標準価格 33,100円(税別)
(リモコン付)

アートツール

画像入力

カラーイメージスキャナ※1
CZ-8NS1
標準価格 188,000円(税別)

スキャナ用パラレルボード
CZ-6BN1
標準価格 29,800円(税別)

映像入力

カラーイメージユニット※2
CZ-6VT1-BK
CZ-6VT1
標準価格 69,800円(税別)

映像出力

ビデオボード※3
CZ-6BV1
標準価格 21,000円(税別)

プリンタ

カラープリンタ

24ドット
熱転写カラー漢字プリンタ
★CZ-8PC3
標準価格 65,800円(税別)
(信号ケーブル同梱)

48ドット
熱転写カラー漢字プリンタ
CZ-8PC4
CZ-8PC4-GY
標準価格 99,800円(税別)
(信号ケーブル同梱)

ドットプリンタ

24ピン
カラー漢字プリンタ(80桁)
CZ-8PG1
標準価格 130,000円(税別)
(信号ケーブル同梱)

24ピン
カラー漢字プリンタ(136桁)
CZ-8PG2
標準価格 160,000円(税別)
(信号ケーブル同梱)

カラービデオプリンタ

カラービデオプリンタ
CZ-6PV1
標準価格 198,000円(税別)
(信号ケーブル同梱)

24ピン漢字プリンタ(136桁)
CZ-8PK10
標準価格 97,800円(税別)
(信号ケーブル同梱)

カラーイメージジェット

カラーイメージジェット※4
IO-735X
標準価格 248,000円(税別)
(信号ケーブル別売)

ファイル

光磁気ディスク

光磁気ディスクユニット※5
(594MB)
CZ-6MO1
標準価格 450,000円(税別)
(SCSIケーブル同梱)
※光磁気ディスクカートリッジは別売です。別売のJY-701MPA 標準価格30,000円(税別)をご使用ください。

ハードディスク

ハードディスクユニット(20MB)
CZ-620H
標準価格 178,000円(税別)

増設用ハードディスクドライブ(40MB)
(CZ-602C/603C/652C/653C内蔵用)
CZ-64H
標準価格 120,000円(税別)
(取付費別)
※取付に関してはシャープお客様ご相談窓口にてご相談ください。

※1 ご使用に際しては、カラーイメージスキャナCZ-8NS1に同梱のRS-232Cケーブルで接続するか、より高速のパラレルデータ伝送を行う場合、別売のスキャナ用パラレルボードCZ-6BN1標準価格29,800円(税別)で接続してください。
※2 CZ-603D 604D、CU-21HDをご使用の場合は、RGBシステムチューナーCZ-6TU(別売)が必要です。 ※3 ビデオ出力は15.75kHzテレビ標準信号です。また、拡張I/Oスロットは2スロット使用します。
※4 別売の信号ケーブルIO-73CX標準価格5,500円(税別)で接続して下さい。 ※5 CZ-600C、601C、602C、603C、611C、612C、613C、652C、653C、662C、663Cに使用の場合は、別売のSCSIボード(CZ-6BS1)が必要です。(但し、CZ-623Cは不要)また、X68000用OS Human68K ver.2.0以上にてご使用ください。(光磁気ディスクカートリッジは別売のJY-701MPA標準価格30,000円(税別)をご使用ください。) ※6 ご使用に際しては、あらかじめ別売の1MB増設RAMボードCZ-6BE1 標準価格

X1・X1 turbo シリーズ用 周辺機器

標準価格は税別です。

カラーディスプレイ		
●21型カラーディスプレイ※1	CU-21HD	148,000円

映像・画像入力編集装置		
●カラーイメージスキャナ	CZ-8NS1	188,000円
●カラーイメージボードII	CZ-8BV2	39,800円
●立体映像セット	★CZ-8BR1	29,800円
●パーソナルテロッパ※2	CZ-8DT2	44,800円

FM音源		
●ステレオタイプFM音源ボード	CZ-8BS1	23,800円

スピーカー(2本1組)標準装備、ミュージックツール同梱

プリンタ		
●24ピンカラー漢字プリンタ(80桁)	CZ-8PG1	130,000円
●24ピンカラー漢字プリンタ(136桁)	CZ-8PG2	160,000円
●24ピン漢字プリンタ(136桁)	CZ-8PK10	97,800円
●24ドット熱転写カラー漢字プリンタ	★CZ-8PC3	65,800円
●48ドット熱転写カラー漢字プリンタ	CZ-8PC4	99,800円
●48ドット熱転写カラー漢字プリンタ	CZ-8PC4-GY	99,800円
●カラービデオプリンタ	CZ-6PV1	198,000円
●カラーイメージジェット	IO-735X	248,000円

ファイル		
●ミニフロッピーディスクユニット(2HD・2D)※3	★CZ-520F	118,000円

シャープ株式会社
●お問い合わせは…シャープ㈱電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)

資料請求券 X68000ペリフェラル Oh!X 12体

お望みのパワーシステムへ。

シャープペリフェラルファミリー X68000

ボード

拡張メモリ

1MB増設RAMボード
(CZ-600C専用)
CZ-6BE1
標準価格 35,000円(税別)

1MB増設RAMボード
(CZ-601C/611C/652C/653C/662C/663C用)
CZ-6BE1B
標準価格 28,000円(税別)

2MB増設RAMボード ※6
CZ-6BE2
標準価格 79,800円(税別)

4MB増設RAMボード ※6
CZ-6BE4
標準価格 138,000円(税別)

インターフェイス

ユニバーサルI/Oボード
CZ-6BU1
標準価格 39,800円(税別)

GP-IBボード
CZ-6BG1
標準価格 59,800円(税別)

増設用RS-232Cボード
(2チャンネル)
CZ-6BF1
標準価格 49,800円(税別)

NEW

SCSIボード ※7
CZ-6BS1
標準価格 29,800円(税別)
(ソフトウェア(SCSIユーティリティ)同梱)

数値演算プロセッサ

数値演算プロセッサボード
CZ-6BP1
標準価格 79,800円(税別)

FAX

FAXボード
CZ-6BC1
標準価格 79,800円(税別)

MIDI

MIDIボード
CZ-6BM1
標準価格 26,800円(税別)

ネットワーク

モデム

モデムユニット ※8
CZ-8TM2
標準価格 49,800円(税別)
(RS-232Cケーブル同梱)

RS-232Cケーブル

RS-232Cケーブル
(平行接続型)
CZ-8LM1
標準価格 7,200円(税別)

RS-232Cケーブル
(クロス接続型)
CZ-8LM2
標準価格 7,200円(税別)

LANボード

LANボード
CZ-6BL1
標準価格268,000円(税別)
(イーサネット用)
CZ-6BL2 NEW
標準価格298,000円(税別)
(イーサネット/チーパネット両用)
※電源ユニット・ソフトウェア
(ネットワークドライバVer1.0)同梱

入力

インテリジェントコントローラ
CZ-8NJ2
標準価格 23,800円(税別)

マウス・トラックボール
CZ-8NM3
標準価格9,800円(税別)

トラックボール
CZ-8NT1
標準価格 13,800円(税別)

マウス
CZ-8NM2A
標準価格 6,800円(税別)

ジョイカード
CZ-8NJ1
標準価格 1,700円(税別)

その他

拡張スロット

拡張 I/Oボックス(4スロット)
(CZ-600C/601C/602C/603C/611C/612C/613C/623C用)
CZ-6EB1-BK
CZ-6EB1
標準価格 88,000円(税別)

スピーカー

アンプ内蔵
スピーカーシステム(2本1組)
AN-S100
標準価格 36,600円(税別)

システムラック

システムラック
(CZ-600C/601C/602C/603C/611C/612C/613C/623C用)
CZ-6SD1
標準価格 44,800円(税別)

35,000円(税別・CZ-600C用)、CZ-6BE1B 標準価格28,000円(税別・CZ-601C、CZ-611C、652C、653C、662C、663C用)を増設してください。 ※7 CZ-600C、601C、602C、603C、611C、612C、613Cに装着の場合、I/Oスロット2に装着ください。CZ-652C、653C、662C、663Cに装着の場合はI/Oスロット4に装着ください。また、CZ-6BG1、6BU1、6BL1、6BL2、6BN1などのボードは、接続コネクタとの関係で本ボードとの併用はできませんのでご注意ください。なお、本ボードはX68000用OS Human 68K ver.2.0以上にてご使用ください。 ※8 モデムユニットCZ-8TM2に同梱のソフトはX1/X1ターボシリーズ用です。

品名	品番	価格
●ミニフロッピーディスクユニット(2D)	★CZ-502F	99,800円
●ミニフロッピーディスクユニット(2D・1ドライブ)	CZ-503F	49,800円
●増設用ミニフロッピーディスクドライブ(2D)※4	CZ-53F-BK	19,800円

拡張ボード・その他

品名	品番	価格
●モデムユニット(300/1200ボー)	CZ-8TM2	49,800円
●320KB外部メモリ	CZ-8BE2	29,800円
●RS-232C・マウスボード ※5	CZ-8BM2	19,800円
●フロッピーディスクインターフェイス※6	CZ-8BF1	14,800円
●JIS第1水準漢字ROM ※7	CZ-8BK2	19,800円
●RS-232C用ケーブル(平行接続型)	CZ-8LM1	7,200円
●RS-232C用ケーブル(クロス接続型)	CZ-8LM2	7,200円
●拡張 I/Oボックス	CZ-8EB3	33,800円
●RFコンバータ ※8	AN-58C	2,980円
●インテリジェントコントローラ	CZ-8NJ2	23,800円
●マウス・トラックボール	CZ-8NM3	9,800円
●マウス	CZ-8NM2A	6,800円
●トラックボール	CZ-8NT1	13,800円
●ジョイカード	CZ-8NJ1	1,700円
●チルトスタンド	CZ-6ST1-E・-B	5,800円
●高性能CRTフィルター ※9	BF-68PRO	19,800円
●スキャナ用パラレルボード ※10	CZ-8BN1	27,800円

●品番中の-表示は、B〈ブラック〉・E〈オフィスグレー〉を示します。※1 X1ターボZシリーズ用 ※2 CZ-862Cには接続できません ※3 X1ターボシリーズ用 ※4 CZ-830C用 ※5 X1シリーズ用 ※6 CZ-850CでCZ-520Fを使用する場合に必要 ※7 CZ-800C、801C、802C、803C、811C、820C用 ※8 CZ-820C、822C、830C用 ※9 14/15型用 ※10 CZ-8NS1用 ●接続等の説明につきましては、周辺機器総合カタログをご参照ください。

★印の商品は在庫僅少です。

電子機器事業本部液晶映像システム事業部第2商品企画部 〒162東京都新宿区市谷八幡町8番地☎(03)260-1161(大代表)

SHARP

ハイアビリティを実証する多彩なソフトウェア。

ドロー編集、WYSIWYG印刷、こんなC.G.ツールが欲しかった。

本格的なロゴタイプやPOPを簡単に作成できるグラフィックツールです。優先順位が任意に指定できるドローセル、ペイントセル、テキストセルの3つの仮想セルで、目的にあった自由なグラフィックが駆使できます。また印刷は、画面イメージがそのまま印刷イメージとなるWYSIWYG（What You See Is What You Get）を実現。A6/A5/A4/A3/B6/B5/B4/葉書サイズで8色カラー印字できます。

〈ドローセル〉ベジェ曲線によって少ないデータ量でも複雑な絵を描くことができます。エンベロープ変形を始めとした豊富な編集機能を持っており、拡大、縮小しても絵の美しさは変わりません。またテキストセルで作成したベクトルフォントデータを自由に変形し、オリジナルロゴタイプやPOPを作成できます。

〈ペイントセル〉ペンやエアーブラシ、ペンキなどを使って、ピクセルで構成されたビットマップ図形を描くことができます。また、「NEW PrintShop PRO-68K」や「X-BASIC」、「Z's STAFF PRO-68K」のデータ取り込みやイメージスキャナによる取り込みをサポートしています。

〈テキストセル〉通常の文字入力機能に加え、ベースライン変形などの多彩な編集機能によって自由に文字の加工ができます。また英数文字のベクトルフォントを標準装備。さらに「Z's STAFF PRO-68K Ver2.0」、「書体倶楽部」の日本語ベクトルフォントが利用可能。また、内蔵の漢字ROMフォントも自動的にベクトルフォントデータに変換しますので、簡単に日本語ロゴタイプを作成することができます。

※「Z's STAFF PRO-68K」、「書体倶楽部」は、㈱Zeitの製品です。
※本ソフトの動作には、メインメモリ2MBが必要です。

CANVAS PRO-68K CZ-249GS 標準価格29,800円（税別）

● 主として個人用のさまざまなジャンルのデータが収められているドローグラフィックデータ集です。

海のデータ/動物のデータ/スポーツのデータ/鳥のデータ/人物のデータ/食物のデータ/昆虫のデータ

CANVAS PRO-68K ドローグラフィックライブラリ VOL.1
CZ-255GS 標準価格8,800円（税別）

● 主としてビジネス用のさまざまなジャンルのデータが収められているドローグラフィックデータ集です。

OA関係のデータ/飾りのデータ/コンピュータ関連のデータ/POPのデータ/国旗のデータ/字体のデータ/地図のデータ/乗り物のデータ

CANVAS PRO-68K ドローグラフィックライブラリ VOL.2
CZ-256GS 標準価格8,800円（税別）

ソースコードデバッガをはじめ、各種開発ツールを強化。バージョンアップされたCコンパイラ。

Cのソースレベルでデバッグできる「ソースコードデバッガ」を搭載したほか、各種開発ツールを強化した総合開発ツールです。また、ライブラリはHuman 68k ver2.0の拡張DOSコールもサポートしているなど、よりX68000のハードウェアを活かせる豊富なライブラリ（830種以上）となっています。C言語の標準であるANSI規格準拠をさらに強化。「プログラム保守ユーティリティ（MAKE）」や「ライブラリアン」など各種ツールを追加しました。その他「BASIC-Cコンバータ」、「アセンブラ」、「リンカ」、「デバッガ」、「ソースコードデバッガ」、「アーカイバ」、「コンバータ」、などのツールが装備されています。

※C compiler PRO-68K（CZ-211LS）を既にお持ちの方は、登録カードをもとに有償バージョンアップを行います。
※本ソフトの動作にはメインメモリ2MBが必要です。

CZ-245LS
標準価格44,800円（税別）

トラブルエラーの悩み解消！「XBAStoC」の強力ツールの登場です。

X-BASICプログラムのコンパイル時、発見しづらいトラブルエラーに悩まされていたプログラムの問題点をひとつひとつ指摘。エラーとなる直接原因だけでなく、注意項目も指摘します。これにより、X-BASICでは実行できたのにコンパイルするとエラーが発生する、といったプログラムの修正が簡単にできます。

● 指摘したトラブルの結果を、画面やプリンタなどの外部デバイスに簡単に出力できます。● エラーラインとエラーレポート、2つのエラーファイルを自動的に生成。● グラフィカルな画面による簡単操作。● コマンドラインからダイレクトに操作を指定。バッチファイルに組み込むなどの修正作業の自動化が可能。● GP-IBボード（CZ-6BG1）とユニバーサルI/Oボード（CZ-6BU1）付属の拡張外部関数もコンパイル可能。

※X-BASICプログラムをコンパイルするためには、別売の「C compiler PRO-68K」（CZ-211LS）または「C compiler PRO-68K ver2.0」（CZ-245LS）が必要です。

XBAStoC CHECKER PRO-68K
CZ-260LS 標準価格9,800円（税別）

シャープ株式会社 ● お問い合わせは……シャープ㈱電子機器事業本部液晶映像システム事業部第2商品企画部 〒162 東京都新宿区市谷八幡町8番地 ☎(03)260-1161（大代表）へ。

シャープオリジナルソフトウェア
X68000

お望みのワークベンチへ。

ビジネスツール

Hyperword
■CZ-251BS 標準価格39,800円(税別)

X68000の優れたグラフィック環境を活用し効率的に文書を作成するためのインテリジェントワープロです。アイデアプロセッサ機能、ハイパーテキスト機能などをサポート。データの整理やプレゼンテーションツールなど幅広い用途に利用できます。

CYBERNOTE PRO-68K
■CZ-243BS 標準価格19,800円(税別)

プライベートなデータやビジネスデータを簡単な操作で管理・運営できるパーソナルデータベースです。リフィル、タックシール、ハガキなどへの印字もOK。シャープ電子手帳とのデータ交換可能(別売の通信ケーブルCE-200Lが必要)。

Stationery PRO-68K
■CZ-240BS 標準価格14,800円(税別)

他のソフトを起動する前に、このStationery PRO-68Kを一度起動するだけで、他のソフトを実行中にも「スケジュール」「住所録」など多彩な機能をワンタッチで使用できます。シャープ電子手帳とのデータ送受信も実現。(別売の通信ケーブルCE-200Lが必要)。

TOP給与計算エキスパート
■CZ-228BS標準価格200,000円(税別)

給与計算から明細発行までを、リアルイメージ入力により自動的に、素早く処理することができます。

TOP財務会計
■CZ-227BS標準価格200,000円(税別)

会計エキスパートシステムとデータベースを搭載し、機能と操作性を両立させた財務会計ソフト。

CARD PRO-68K
■CZ-226BS 標準価格29,800円(税別)

自由なレイアウト画面で入力できるワープロ機能を装備したカード型リレーショナルデータベース。

CARD PRO-68K用システム手帳リフィル集
■CZ-241BS 標準価格9,800円(税別)

CARD PRO-68K用活用フォーム集
■CZ-242BS 標準価格9,800円(税別)

DATA PRO-68K
■CZ-220BS 標準価格58,000円(税別)

入力の手間を軽減するヒストリー機能を装備した、コマンド型リレーショナルデータベースです。

BUSINESS PRO-68K
■CZ-212BS 標準価格68,000円(税別)

スプレッドシート(表計算)、データベース、グラフ作成機能を一体化させた統合ビジネスツールです。

アートツール

NEW PrintShop PRO-68K
■CZ-221HS 標準価格19,800円(税別)

オリジナリティあふれるはがき等、簡単に作成、印刷できるホームプロダクティビリティツール。

グラフィックライブラリ VOL.1
■CZ-235GS 標準価格8,800円(税別)

グラフィックライブラリ VOL.2
■CZ-236GS 標準価格8,800円(税別)

通信ツール

Communication PRO-68K ver2.0
■CZ-257CS 標準価格19,800円(税別)

Communication PRO-68Kのバージョンアップ版です。MNPモデムへの対応で、ハードフロー制御(CTS/RTS)をサポートしています。

※バージョンアップ対応中。

開発ツール

SX-WINDOW ver1.0
■CZ-259SS 標準価格6,800円(税別)

複数の作業を同時に処理できる疑似マルチタスクや入出力装置の設定が簡単に行える多機能コントロールパネルを搭載した本格ウィンドウシステムです。IOCSコールを利用したソフトの処理速度を高速化するIOCS.Xを付属。

OS-9/X68000
■CZ-219SS 標準価格29,800円(税別)

マルチタスク機能、リアルタイム機能を活かした使いやすく機能的なOS環境を提供します。

※OS-9はマイクロウェア社の登録商標です。

Human68k ver2.0
■CZ-244SS 標準価格9,800円(税別)

システムパフォーマンスをさらに高める処理機能を付加したHuman68kの最新バージョンです。

THE福袋V2.0
■CZ-224LS 標準価格9,980円(税別)

AI-68K(Staff LISP/OPS PRO-68K)
■CZ-234LS標準価格188,000円(税別)

サウンドツール

Musicstudio PRO-68K ver.1.1
■CZ-252MS 標準価格28,800円(税別)

MUSIC PRO-68K〔MIDI〕
■CZ-247MS 標準価格28,800円(税別)

ソングライブラリ〈101曲集〉
■CZ-248MS 標準価格8,800円(税別)

Sampling PRO-68K
■CZ-215MS 標準価格17,800円(税別)

SOUND PRO-68K
■CZ-214MS 標準価格15,800円(税別)

MUSIC PRO-68K
■CZ-213MS 標準価格18,800円(税別)

ゲーム

シューティングゲーム
〈ツインビー〉
■CZ-217AS
標準価格7,800円(税別)
©KONAMI 1988

シューティングゲーム
〈沙羅曼蛇〉
■CZ-218AS
標準価格8,800円(税別)
©KONAMI 1989

ブロックゲーム
〈アルカノイド〉
■CZ-222AS
標準価格7,800円(税別)
©TAITO CORP. 1987

ドライブゲーム
〈フルスロットル〉
■CZ-231AS
標準価格8,800円(税別)
©TAITO CORP. 1988

スポーツゲーム
〈熱血高校ドッジボール部〉
■CZ-232AS
標準価格7,800円(税別)
©TECHNOS JAPAN CORP. 1988

アクションゲーム
〈パックマニア〉
■CZ-233AS
標準価格7,800円(税別)
©NAMCO

アクションゲーム
〈ニュージーランドストーリー〉
■CZ-230AS
標準価格8,800円(税別)
©TAITO CORP. 1989

スポーツゲーム
〈V'BALL〉
■CZ-246AS
標準価格7,900円(税別)
©TECHNOS JAPAN CORP. 1989

バイクレーシングゲーム
〈スーパーハングオン〉
■CZ-238AS
標準価格8,800円(税別)
©SEGA 1987

ジェットヘリ・シミュレーションゲーム
〈サンダーブレード〉
■CZ-239AS
標準価格9,500円(税別)
©SEGA 1987

アクションゲーム
〈ダウンタウン熱血物語〉
■CZ-254AS
標準価格8,800円(税別)
© TECHNOS JAPAN CORP. 1989

アクションゲーム
〈サイバリオン〉
■CZ-229AS
標準価格8,800円(税別)
©TAITO CORP. 1988

スポーツゲーム
〈熱血高校ドッジボール部サッカー編〉
■CZ-262AS
標準価格8,800円(税別)
©TECHNOS JAPAN CORP. 1990

LUCY SHOT
ルーシーショット

ソフテック発、ゲームソフト第2弾、エキサイティング・ピンボールゲーム!

大人のゲーム「ピンボール」には、アメコミがよく似合う。

リアルな仕掛けがその気にさせる、本格的ピンボールゲーム第2弾がいよいよ登場! 美女ルーシーをはじめ、キャラ、グラフィックなど、画面いっぱい、まんまアメリカン・コミックの世界だ! マニアックな設定やマシンの動きは、キミに、ゲーセンを超えた興奮を味あわせてくれる。アウトローを気取って、ジーンズに皮ジャンで挑戦してくれ! **11月16日新発売 定価7,800円(税別)**

対応機種:PC-9801VM以降(5″2HD) PC-9801UV以降(3.5″2HD)
X68000 ※要アナログディスプレイ・FM音源対応、要16色ボード

●物理的にシミュレートした、なめらかなボールの動き●迫力満点アメコミ調グラフィック●フィールドチェンジ、プランジャー、4フリッパー、キックアウトホール、スロットマシン、ロールオーバーレーン、バンパー、チルトetcの趣向を凝らした仕掛けの数々。

※画面写真はX68000のものです。

PINBALL PINBALL

「ピンボール・ピンボール」好評発売中!
定価7,800円(税別)
対応機種:
PC-9801VM以降(5″2HD)
PC-9801UV以降(3.5″2HD)
X68000

※要アナログディスプレイ・FM音源対応、要16色ボード

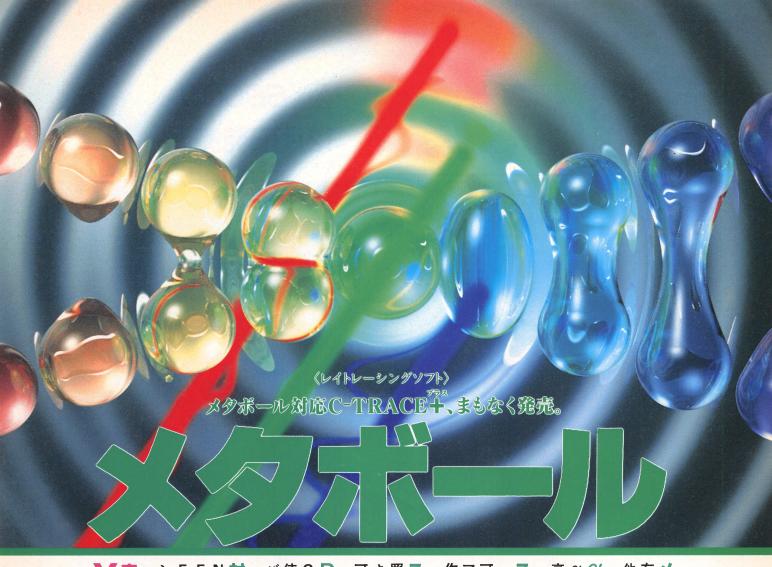
eXOn

ライフ・シミュレーション
「エグゾン」
来春3月発売予定!

全国通信販売 通信販売ご希望の方は、商品名、機種名、メディア名、住所、氏名、電話番号を明記の上、定価に消費税(3%)をプラスして、現金書留で下記までお申し込みください。(送料無料)

SOFTEC

プログラマー(PC98、X68等)、グラフィック、ミュージック、企画のスタッフを募集中です。ご連絡ください。

〒191 東京都日野市日野1161-1 カトレアマンション102
日本ソフテック株式会社 (ユーザーサポート係)
TEL.0425-82-1502 FAX.0425-87-3991

〈レイトレーシングソフト〉

メタボール対応C-TRACE＋(プラス)、まもなく発売。

メタボール

メタボール…
有機的な質感を表現する事ができます。
他のプリミティブと論理演算ができます。

αチャンネル機能…
αチャンネルへの対応により、
高度な合成作業が可能となります。

スコープ機能…
一度生成した画像の部分修正が
可能となります。
アニメーションへの応用により、
作業の大幅な効率化が図れます。

スポット光源への対応…
照射範囲の設定が自由にできます。
さらに照射範囲の境界のぼかしも
可能です。

DOS-EXTENDERへの対応…
CPUがV30のユーザーの方も
使用できるようにノーマル版も
バンドリング。(98版のみ)

対応機種…
NEC PC9801シリーズ
EPSON PC286シリーズ
EPSON PC386シリーズ
シャープ X68000シリーズ

定価…
¥198,000

レイトレーシングを高速にしたい方へ。

C-TRACE TP Ver.3.0 ¥298,000

トランスピュータボード（T-800×1+4M）＋
C-TRACE Ver.3.0トランスピュータ版ソフトウェア〈ソフト＋ハード〉
68000＋C-TRACE Ver.3.0のスピードの〝約170倍〟
80386＋C-TRACE Ver.3.0のスピードの〝約40倍〟
●対応機種／98版…ＰＣ9801シリーズ、または互換機なら新旧問わずほとんどの機種に対応。ただし、標準拡張スロットがない機種には、装着できません。68版…X 68000全機種
★もっとスピードを上げたい方へ…並列処理によってスピードアップ可能！
増設トランスピュータモジュール（1ＴＲＡＭ）¥298,000

★フルカラーフレームバッファ	¥69,800
C-TRACE98 EXTENDER	¥128,000
C-TRACE98 Ver. 3.0	¥98,000
C-TRACE68 Ver. 3.0	¥98,000
C-TRACE TOWNS	¥68,000
C-TRACE NEWS Ver. 3.0	¥530,000
★C-TRACE98 TP Ver. 3.0	¥298,000
★C-TRACE68 TP Ver. 3.0	¥298,000

標示価格に消費税は含みません。★の製品は店頭販売いたしておりません。直接当社まで、お申し込みください。
クレジット可

バージョンアップ受付中。

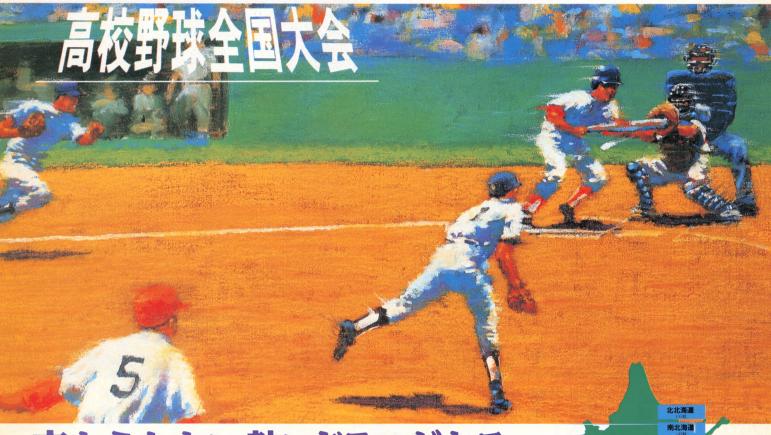
●お問い合せ先　株式会社キャスト
〒158 東京都世田谷区等々力2-1-13
TEL.03-705-1065 FAX.03-705-5224

ARTDINK

●5"2HD（3枚組）

12月7日発売。

標準価格　9,500円

高校野球全国大会

忘れられない熱いドラマがある。

陽が落ちるまで、夢中で千本ノックを続けたあの日の汗。初めてライバルを制した、あの練習試合の涙。
そして、厳しい予選をくぐりぬけ、きょう、夢にまで見た甲子園の土を踏んだ。
さあ、球児たちの燃える視線が、あなたの采配を待っている。
彼らに栄光の優勝旗を抱かせることができるだろうか?!
感動の高校野球シミュレーションゲーム、「栄冠は君に」。
練習で選手を育て、試合で作戦を練り、約4000校のトーナメントを勝ち抜く……
監督としてのあなたの手腕が、忘れられない夏のドラマを誕生させる！

全国3,990校の頂点を極めるのは、君だ。

＊3990校とは第71回全国高校野球選手権大会の参加総数です。
本ゲームも3990校登場します。

株式会社 アートディンク　〒275 千葉県習志野市津田沼2 11 20
TEL 0474 77 7541（ユーザサポート専用）

●お買い求めは、全国パソコンショップにて。
●通信販売（送料無料）をご希望の方は、住所・氏名・電話番号・商品名・機種名・メディア名を明記して左記まで消費税3%を同封の上現金書留にてお申し込み下さい。

（表示価格に消費税は含みません）

NEW CONCEPT CREATIVE TOOL

G・TOOL

G•ツール

FOR X68000

アートせよ。

彩先端のアートキャンバスに彩り鮮やかに感性を咲かせてください。ザインのG＝ツールは、単なるペイントツールにとどまらず、ゲームデザインをはじめとしたひとつの作品を創造する上で必要不可欠なグラフィック・キャラクタ・背景作成のすべてを備えた新感覚のグラフィックツール。驚くほどの自由さと、繊細なクリエイターのこだわりにまでアプローチしたそのコンセプトは、あなたの感性を刺激せずにおきません。

■ 概要

G•ツール	
GR-EDITモード	マルチウインドウシステム 最高12枚まで描画ウインドウが開けます。
	ユーザーアイコンシステム 使い勝手に合わせて、自分流のアイコンボードが作成可能。
	マウス機能定義システム マウスの左右に機能定義が可能。
	高速メニューウインドウ処理 メニューウインドウの開閉が瞬時に行えます。
BG-EDITモード	スプライト処理 作成から修正、アトリビュートが行えます。
	スプライトカラー処理 16ページ分まとめて処理できます。
	背景の作成 最大250画面分を自由に設定することができます。
	キャラクタチェック機能 単独チェックのほか、背景と重ねてのチェックも可能。

好評発売中

■ 5"2HD
定価¥28,000(税別)

zainsoft

株式会社 ザインソフト 〒651 神戸市中央区磯辺通2丁目2-10新南泰ビル10F TEL.(078)242-2855(代表)

資料・オートデモ 請求券 OH!X 12月号

Humming Bird Soft™
株式会社エム・エー・シー ハミングバードソフト
〒530 大阪市北区曽根崎2丁目2番15号

怪魔退散!!

踏み出すとそこは殺戮の地。
強剛者達よ、立ち上がれ!!

neXt
RPG・ACT・SLG、最強のラインナップで次世代体験……neXt!

UNDEADLINE

幻獣鬼™

けんじゅうき

X68000

◀5"2HD・3枚組 標準価格¥7,800(税別)▶

ACT-neXt

はじめからひとつひとつステージをクリアしていくのが、ほとんどのアクションゲームの形態。クリアするたびに難度があがってなかなか先のステージにチャレンジできない。そこであきらめてしまったり、「だから、プレイしたくないんだ、苦手なんだ」という人もいるだろう。この課題に取り組み、解決したのが、ステージ・セレクション・システム、略してS・S・S。8ステージのどのステージからでもプレイ可能。好きなように選択して楽しめる。
だからといって、初心者のゲームと勘違いしては困る。「アクションゲームは正攻法にプレイすべきだ」と思っている人には、難度を非常に高く設定した「HARD」を用意している。どのくらいのゲーマーが最後まで到達できるか、楽しみだ。

★プレイヤーは、戦士、魔道士、忍者の3キャラクターから選択。
　ゲーム途中で一度クラスチェンジが可能。
★各キャラクターは、通常武器の他にオプション武器が3種類装着可能。
★ジョイスティック対応。
★FM音源とADPCMに対応。
★ゲーム中の全曲が聴けるミュージックモードあり。

RPG-neXt
ルーンワース 黒衣の貴公子

- X68000
- PC-9801VM、UVシリーズ
  PC-286、386シリーズ、NOTE対応
- PC-8801SRシリーズ・VA、98DO対応
- MSX2/MSX2+

標準価格 各¥8,800(税別)

NEW 3D GOLF SIMULATION

オーガスタ・ナショナル・ゴルフ・クラブと正式契約

SLG-neXt
遙かなるオーガスタ

■通信販売をご希望の方は、現金書留で料金と商品名・機種名と電話番号を明記の上、当社宛にお送りください。(速達希望の方は300円プラス)

Phone052-776-8500

株式会社 ティーアンドイーソフト
〒465 名古屋市名東区豊が丘1810 PHONE:052-773-7770

KONAMI
※この画面は開発中のものです。
PARODIUS
パロディウスだ!TM
―神話からお笑いへ―
© KONAMI 1990

笑って砕けよ。
あぁぁぁ〜っ、かくも恐ろしき世界よ。
二度と、私の知性を乱してくれるなとあれほど頼んだのに、
再びその滑稽な姿かたちを現したな。
しかも、ゲームセンターで我が物顔をしていた
そのまんまの容姿ではないか。
ここまでやってしまったのか、というパロディウス倫理委員会の
ささやきを縦横無尽に交わしながら、
ほぼ完全なる移植にむけ着々と進攻している。
X68だから、よけいにこわい。
もうプレイヤーは体当たりで砕けちるしかないのだろうか。
X68000シリーズ
コナミ株式会社

リアリティ。

※この画面は開発中のものです。

宇宙は、野望だけでは支配できない。

宇宙暦796年、銀河系はゴールデンバウム王朝が支配する銀河帝国と、その専制政治に反対する自由惑星同盟の両陣営が激しい戦闘を繰り返していた……。圧倒的支持を得た「銀河英雄伝説」を遥かに凌ぐスケールで、今新たな伝説が生まれようとしている。銀河英雄伝説IIだ。帝国軍の若き天才ラインハルト、そしてヤン・ウェンリーの熱い闘いが、再び始まる。星系マップは従来の4倍、3Dグラフィックによる戦闘シーンなど、あらゆる面でパワーアップされている。田中芳樹原作の大人気スペースオペラ「銀河英雄伝説」。宇宙の歴史を変える闘いは、ここに始ろうとしている。

SPACE WAR SIMULATION

銀河英雄伝説II

銀河英雄伝説II　X68000シリーズ 11月30日発売　¥9,800（税別）

●5 2HD（4枚組）●X68000専用グラフィック●2重スクロール●MIDI音源対応
●FM音源・ADPCM対応●HMS（HYPER MOUSE SYSTEM）搭載

BOTHTEC ボーステック・ソフトウェア
株式会社クエスト（旧ボーステック株式会社）〒158 東京都世田谷区用賀2-18-8 TEL.03-708-4711

スタッフ募集　ゲーム企画・プログラマー・音楽担当者
採用係連絡先　TEL.03-708-4712

※通信販売(送料サービス):品名・機種・住所・氏名・電話番号を明記して、
現金書留でお申し込みください。
画面は開発中のものです

勇者たちよ！いま一度力（パワー）を…

新たな挑戦の幕開けが迫る！

■世界をそして日本を興奮の渦に巻き込んだリアルタイムRPG「ダンジョンマスター」。その続篇が帰ってきた！「ロード・カオス」を倒し世界に平和と秩序をもたらした勇者達…。しかし「ロード・カオス」は生きていた。あの時、彼は既に自分が勇者達に倒されることを知り、秘かに新しいダンジョンを作って悪の力を蓄えていたのだ。

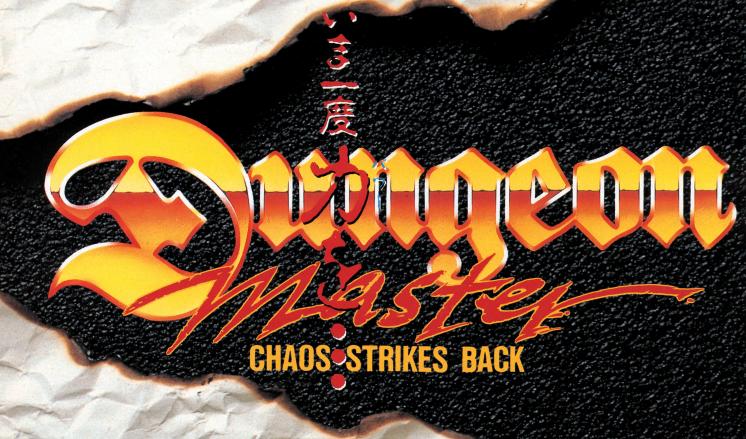
■アニメを使ったイントロダクション、FM音源対応による音楽とサウンド、チャンピオンの顔を自由に書き換えることのできるキャラクターエディット機能、さらには日本版独自のプレイヤーの現在地を一目で表示するマップ表示機能や新しい魔法の追加など熱中度さらにパワーアップ。

■前作で育てた勇者をそのまま使って冒険に旅立つもよし、新たに用意された勇者を編成してダンジョンに向かうもよし、ドキドキしながらこの興奮を味わって下さい。前作以上の難しさとおもしろさだけは保証します。

Dungeon Master

CHAOS STRIKES BACK

戦 い は 終 わ ら な い 。

いよいよ発売！

続 ダンジョン・マスター｜カオスの逆襲

■12月中旬発売 X68000版　¥9,800（税抜）

〈注：24KHzモード対応モニターならフル画面表示〉

発売：ビクター音楽産業株式会社

ストーリィ性を持ったドラマティックなゲーム展開

シュヴァルツシルトの最大の特徴は、そのゲームシステムにあります。単なるウォーシミュレーションではなく、ゲームを進めていくにしたがって、次々に新たな目的が現われ、プレイヤーは知らず知らずにゲームのシナリオに引き込まれていくという、ドラマティックなゲーム展開が魅力の、ＳＦシミュレーションゲームです。

究極のゲームシナリオ

ゲームのおもしろさはシナリオで決まります。軍事行動、外交政策調査・研究、資金運用、商業取り引きといった戦略要素を完璧にシミュレート。シミュレーションゲームの面白さを徹底的に追求した究極のゲームシナリオです。

SCENARIO SIMULATION GAME

狂嵐の銀河

Schwarzschild

シュヴァルツシルト

●5"2HD・2枚組￥12,800(価格には消費税は含まれておりません)

MIDI対応

全方向スクロール、ロボットシューティングの極限!

アーケード版アトミックロボキッド、
X68000に登場!
'90年年末、ついにその全貌を現わす。

ATOMIC ROBO-KID アトミック・ロボキッド

12月
発売予定

2XXX年、核戦争の難を逃れた少数の人々は、地下に隠れて放射能の嵐がやむのを待ち、地上に出た。しかし、人々のDNA(染色体)は、わずかに残った放射能によって破壊され、人類は自分達の子孫を残せなくなってしまう。トミタ博士は、わずかに残された人々をシェルターに冷凍保存しDNA正常化のプログラムを開発する。ところが、シェルターに向かうロボキッドが動き出す直前に博士は、死んだ。自分の目的も解らずに、目覚めたロボキッド。……はたして、ロボキッドは人類を救う事ができるのか?

X68000対応 5"-2HD
●ローランド社
MT-32、CM32L、CM64完全対応
MIDIインターフェイスボードC-Z-6BMI
又は、SACOM製SX-68Mが必要です。
標準価格8,800円

密林深く眠る失せし文明に、
蘇る血の運命(さだめ)。すべての謎は、

魅由の繰り広げるミステリアスアニメーションアドベンチャー第2弾!!

闇の血族

THE PREDESTINED HOMICIDES #2

一人の少女の下に今、
遥かなる真実を紡ぎだす。

1990年6月

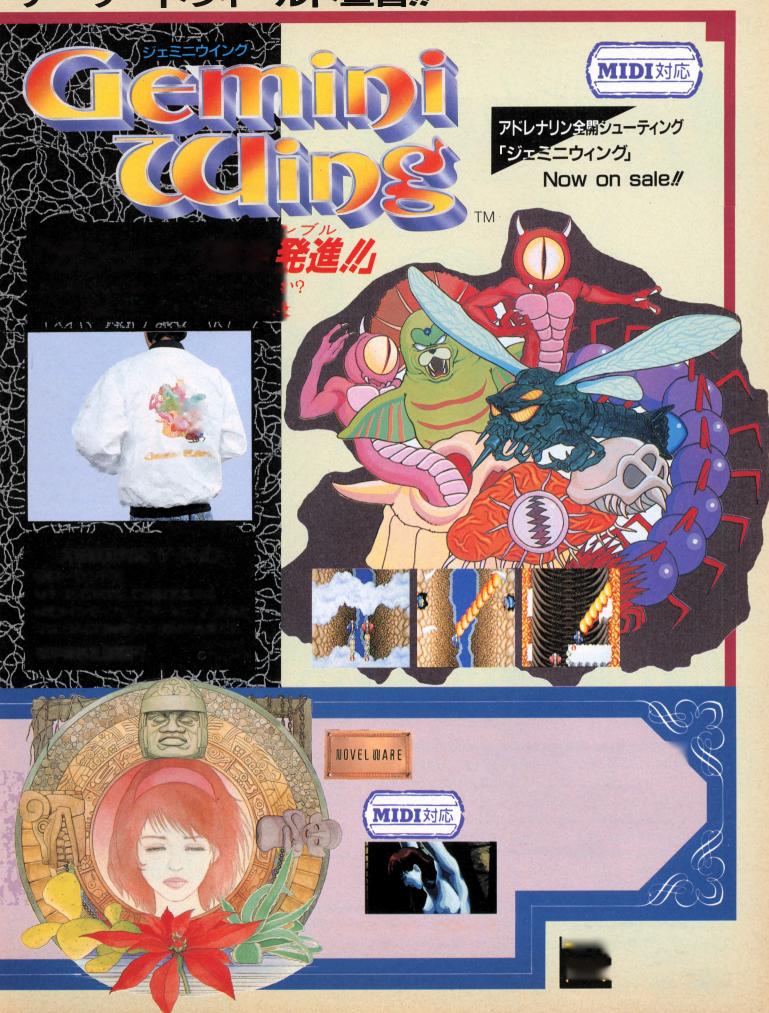
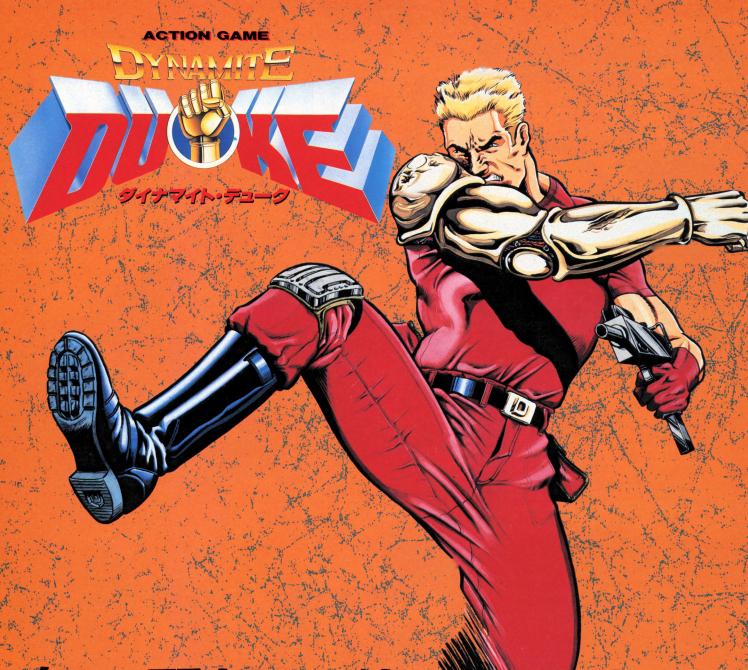
古えの封印は解かれ、時間(とき)の糸車は無数の運命の糸を引き、血染めの歴史絵を紡ぎはじめた。一連の殺人事件に秘められた暗号を解く唯一の手掛かりを求めて、魅由は親友の理沙と共に中米の地へとおもむく。が、そこには思いもよらぬ宿命の罠が待ち受けていたのであった。

完結編

アーケードフィールド宣言!!
ジェミニウイング
Gemini Wing
TM
MIDI対応
アドレナリン全開シューティング
「ジェミニウィング」
Now on sale!!
発進!!
NOVEL WARE
MIDI対応

ACTION GAME

DYNAMITE DUKE

ダイナマイト・デューク

奴の野望は、俺が打ち砕く..!!

西暦2089年、オゾン層の破壊による環境悪化により、人類は滅亡の危機に瀕していた。
この事態を憂慮した各国の指導者は、世界連合軍を設立、厳しい環境下にも耐えうる人体の研究開発に乗りだす。
にバイオテクノロジーの力によって、強化人間を実現させたのである。
しかし、開発に関わった一人の大佐が技術を盗み出し、強化人間をも超える存在を創り上げることに成功。
……、彼の目的は、軍隊を組織し人類を支配することであった!
連合軍は、この野望を打ち砕くため、ひとりの強化人間を送り出した。
機械の右腕を持つ男、陸軍特殊部隊大佐デューク・フリードリッヒ・フェルゼン、コード・ネーム=Red・Dynamite。
しかし、人は彼のことを、こう呼んだ「ダイナマイト・デューク!!」。

●大人気アーケード・ゲームからの移植!●シューティングの興奮+アクションの一体感!●アメリカン・コミック調の美しい画面!●アニメのように派手に動くド迫力デカ・キャラ!●変化に富んだ各ミッションと個性あふれる敵キャラ!●必殺の一撃「ダイナマイト・パンチ」の快感!

'90年11月30日発売.!! X68000専用(ジョイスティック、キーボード対応FM音源対応1PLAYER専用) 5"2HD 3枚組 ¥8,800(税別)

株式会社ヘルツ

STAFF募集!!

〒169 東京都新宿区北新宿2-1-16松本ビル3号2F TEL.03(371)3012

地球を救え!!

緊急事態発生!!

20××年、東西両陣営の対立を背景に、西側ムーンベースの大爆発が起こった。原因究明を急ぐ中、偵察衛生は驚くべき事実を写しだしていた。ムーンベースのマザーコンピュータに寄生する妖しげな植物……。果してエイリアンの侵略なのか。その時、西側の最新鋭戦闘機OF-1『ダイダロス』にスクランブル指令が下った。過酷なシミュレーションファイトをクリアした精鋭達が、東西の壁を越え、地球を、人類を救う戦いへと、今、飛び立とうとしている。

- ■5"2HD2枚組
- ■ジョイステック対応
- ■難易度設定(4段階)可能
- ■フルオート連射機能内蔵

X68000イメージファイト

12月14日発売予定

予価9,700円(税別)

特別付録

X68000版オリジナルテレホンカード

アイレムファンクラブ会員募集中。詳しくは06-534-1060もしくはお近くのテレホンサービスまで。

テレホンサービス　札幌 011-685-9131　名古屋 052-323-0270
福岡 092-475-9400　東京 03-823-5130　大阪 06-535-0651

アイレム株式会社

〒550 大阪市西区阿波座2丁目2番18号 西本町川渕住生ビル13F

RAY-GUN

A riddle of parallel world Roll-Playing game.

画面は98版です

●エルフ初!!フィールドタイプのRPG!!●これまたエルフ初!!アナログ16色を使用したビックなサイズの超美しいグラフィック!!●と、とにかく戦闘シーンを見て、聞いて、感じてください!!一目見たあなたは、エルフがこの作品にかける意欲を強烈に感じるでしょう!!●モンスターの種類も膨大です!!細部まで書き込まれたモンスターを是非、見てください!!●もちろん細部まで書き込まれたのはモンスターだけじゃありません。カワイイ女の子達の(あそこ)から(あそこ)?まで、丹念に書き込まれています!!●今時、ストーリー、アニメーション、BGMがいいのはあたりまえ!!とにかくオープニングだけでも見てやってください!!きっとゲーム大好き人間、つまりあなたの心を揺さぶる事でしょう!!●もちろん、好評のオマケディスクもついています!!

●レイ・ガンは、エルフ初の本格的フィールドタイプのRPGです。スタッフが燃えに燃えて制作しました。
●レイ・ガンでは、これまたエルフ初の「16色アナログ」表示です。女の子の肌はあくまでも美しくきれいに、モンスターは中間色をふんだんに使用し、よりリアルに描かれています。ご期待ください!!
●戦闘シーンも本格的です!!画面だけではわかっていただけないのが残念です。効果音、ビジュアルに気を使いすぎるほど使い、見ているだけでも楽しめるような戦闘シーンを作りました。
●モンスターの種類ももりだくさんです。次にどんなモンスターが出てくるか……ワクワク、ドキドキしながらプレイしてください。
●エルフ自慢のグラフィックがさらにパワーアップ!!16色のアナログはもちろんの事、キャラクターだけでなく、背景にも力を入れて描いています。もちろんメインとなるグラフィックは画面の2/3を占めるビックなサイズです!!
●アニメーションも今回は凝りました……とにかくスタッフ全員で「大きく動かそう!!」をモットーに作り上げました。
●今回も好評のオマケディスクがついています。(オマケディスクとは?=自分が今までクリアーした所までのビジュアル(ちなみに女の子のお楽しみ画面がメインです。)を何度でも手軽に見れるモード)です。
●98、X68000版で3枚組、88版では8枚組(予定)、MSX版では6枚組(予定)というゲームサイズで、お値段がな、なんと6,800円!!……そう、エルフでは一人でも多くのお客様に、(レイ・ガン)を楽しんでいただきたいのです。

X68000版(5インチ[illegible]D)
11月発売予定　定価6,[illegible]00円

ご注目!!
『RAY・GUNオリジナルテレフォンカード』と、とってもファニーな、エルフのロゴ入り「マッチ型電卓」をセットで先着400名の方にプレゼント致します。ご応募の詳細は、製品のマニュアルをご覧ください。

通信販売をご希望の方は…
●現金書留の場合……
商品名、機種、メディアを明記の上エルフまでお送り下さい。
●郵便振替の場合……
郵便局の振替用紙に商品名、機種、メディアを明記の上、口座番号
東京3-191196
エルフまでお申し込み下さい。

株式会社 エ・ル・フ
〒169 東京都新宿区北新宿1-12-5

全国通販

SHARP 認定 PPO-SHOP **O.A.ランド**

(TEL) 03-770-8855

■アフターサービス万全のサポート体制
- 下取・買取は電話で見積りしております。責任を持って下取りさせて頂きます。
- ご注文、お問合せは…。午前10時から午後7時まで
- 商品のお届けは…入金確認後、即日発送致します。

■TEL・FAXのお見積OK!!
■低金利クレジットをご利用下さい。

▶11・15〜12・14

SHARPのことなら なんでもおまかせ!!

ボーナス・シーズン大徳買セール!安く値切ってネ。
お電話下さい。㊙価格をお知らせいたします。

流通事情により、広告表示価格は、お安くなる場合がありますので、ドンドンお電話下さい。

CYBER STICK
■CZ-8NJ2
(定価¥23,800) OAランド特価 ▶¥18,000

MIDIインターフェースボード
サコム(定価¥19,800) OAランド特価
■SX-68M ▶¥15,800
●X68000専用純正コンパチ

SHARP X68000シリーズセット

●次代のインテリジェンス=SX-WINDOW搭載!!

X68000 EXPERTII
- CZ-603C-BK/GY
- CZ-605D-BK/GY
- MD-2HD 20枚

定価合計 ¥453,000

クレジット	12回	¥30,100	24回	¥15,900	36回	¥11,000

X68000 EXPERTII-HD
- CZ-613C-BK/GY
- CZ-605D-BK/GY
- MD-2HD 20枚

定価合計 ¥563,000

クレジット	12回	¥37,400	24回	¥19,800	36回	¥13,700

●SX-WINDOW搭載!!

X68000 PROII
- CZ-653C-BK/GY
- CZ-605D-BK/GY
- MD-2HD 20枚

定価合計 ¥400,000

クレジット	12回	¥26,600	24回	¥14,000	36回	¥9,700

X68000 PROII-HD
- CZ-663C-BK/GY
- CZ-605D-BK/GY
- MD-2HD 20枚

定価合計 ¥510,000

クレジット	12回	¥33,900	24回	¥17,900	36回	¥12,400

●SX-WINDOW搭載!!

X68000 SUPER-HD
- SX-WINDOW搭載
- SCSIインターフェース装備
- 80MBハードディスク搭載
- 3MB大容量メモリ装備
- 高解像度グラフィック

X68000 SUPER-HD
- CZ-623C-TN(チタン)
- CZ-613D-TN(チタン)
- MD-2HD 20枚

定価合計 ¥633,000

クレジット	12回	¥42,100	24回	¥22,300	36回	¥15,400

セットで購入のお客様に、ディスケット(10枚)、ゲームパックサービス中!!
さらに、期間中ゲームソフトが1本付きます。詳しくは、お電話下さい。

周辺機器コーナー

プリンターセットコーナー

①CZ-8PC4(GY)(48ドット/カラー対応/ハガキ可能)
定価¥99,800…… 特価¥64,800

②CZ-8PK10(24ピン漢字プリンター136桁)
定価¥97,800…… 特価¥78,000

③CZ-8PG1(24ピンカラー漢字プリンター80桁)
定価¥130,000…特価¥103,000

④CZ-8PG2(24ピンカラー漢字プリンター136桁)
定価¥160,000…特価¥125,000

OAランド特選品!!

■IO-735X(定価¥248,000)
●カラーイメージジェットプリンター
特価¥190,000

モデム

	機種	価格
オムロン	MD-1200A III	¥14,500
	MD-24FP4 II	¥27,500
	MD-24FP5 II	¥29,800
	MD-24FN4	¥28,000
	MD-24FN5	¥31,300
	MD-24FJ4	¥31,300
	MD-24FJ5	¥34,500
	MD-24FS4	¥28,500
	MD-24FS5	¥34,500
アイワ	PV-A24VM5	¥32,500
	PV-M24	¥28,800
NEC	COMSTAR 2424/4	¥28,800
	COMSTAR 2424/5	¥33,500

X68000用周辺機器コーナー

品名	定価	特価
①CZ-6VT1(カラーイメージユニット)	¥69,800	¥52,500
②CZ-8NS1(カラーイメージスキャナー)	¥88,000	¥141,000
③CZ-6BM1(MIDIボード)	¥26,800	¥20,500
④CZ-8NJ2(インテリジェント・コントローラー)	¥23,800	¥18,000
⑤CZ-6TU(RGBシステムチューナー)	¥33,100	¥25,000
⑥CZ-64H(増設ハードディスク)	¥120,000	¥90,000
⑦CZ-6EB1(拡張I/Oボックス=4スロット)	¥88,000	¥66,000
⑧CZ-6BP1(数値演算プロセッサボード)	¥79,800	¥60,000

■I・O DATA 増設RAMボード

●1MB増設RAMボード PIO-6BE1-A
定価¥25,000
特価¥19,000

●2MB増設RAMボード PIO-6BE2-2M
定価¥50,000
特価¥36,300

●4MB増設RAMボード PIO-6BE4-4M
定価¥88,000

特価¥64,000

SHARP フリートップパーソナルコンピュータ

■AX286N-H2(MZ-8376A)

①Business Mate 標準装備
②20MB・HDD
③フリートップサイズ
④世界標準AX仕様
⑤内部専用スロット
⑥優れた拡張性
⑦SPシステム 標準装備

定価¥398,000
大特価!!
電話で値切ろう!!

電子手帳だよ〜ん便利です!!

①PA-9500 NEW
………▶大特価!!TEL下さい

②PA-8500 台数限定
……▶大特価¥15,000

③PA-7500 台数限定
……▶大特価¥12,000

●数に限りがございますので、お早目にTEL下さい。

OAランド今月の大目玉!!=超A級中古品

◎1年間完全保障、新品同様(美品)=お問い合せ下さい。

■SUPER-HDセット 3セット限り
CZ-623C-TN+CZ-613D-TN(定価¥633,000)……▶特価¥445,000

■EXPERT-IIセット 3セット限り
CZ-603C-BK+CZ-605D-BK(定価¥453,000)……▶特価¥310,000

■PROIIセット 2セット限り
CZ-653C-BK+CZ-605D-BK(定価¥400,000)……▶特価¥275,000

新製品 周辺機器

■光磁気ディスクユニット
●CZ-6MO1
(定価¥450,000)
特価¥360,000

■SCSIボード
●CZ-6BS1
(定価¥29,800)
特価¥24,800

■X BAS to C CHECKER PRO68K
●CZ-260LS(定価¥9,800)
特価¥8,000

OAランド推奨 周辺機器

■SX-WINDOW
(次代インテリジェント・ソフト)
●多機能コントロールパネル搭載の本格ウィンドウシステム。
定価¥6,800
特価¥5,100

■CZ-6BV-1
(ビデオボード)
●ビデオ出力は、テレビ標準信号、拡張I/Oスロット使用

定価¥21,000
特価¥15,800

■CZ-245LS
(C-コンパイラ)
●ソースコードデバッガをはじめ、各種開発ツールを強化。II版
定価¥44,800
特価¥34,000

通信販売のご案内

全国通販

■銀行振込で申し込みの方は商品名及びお客様の住所・氏名・電話番号をお知らせ下さい。
〔振込先〕第一勧業銀行 渋谷支店
普通No.1163457 ㈱オーエーランド

■12月より年内無休!!

■現金書留で送金されるお客様は電話番号と商品名、数量を明記して同封して下さい。■クレジットでご購入を希望される方は申し込み用紙をお送り致しますのでご記入の上返送して下さい。20才以上の方は、原則として保証人不要です。クレジットは1〜60回払で月々5,000円よりご自由に設定できます。

クレジット表

3回	3%	6回	4%	10回	5.5%	12回	5.5%	15回	8%	18回	10%	20回	11%
24回	11.5%	30回	15.5%	36回	16%	42回	20.5%	48回	21%	54回	26.5%	60回	27%

㈱オーエーランド
〒150東京都渋谷区円山町20-4 第5日新ビル1F
☎(03)770-8855 FAX(03)770-7080

関東エリアの送料は、1個につき¥1,000です。
★全商品保証書付。専門のアドバイザーが、お客様のニーズに対応します。
★初期不良・輸送トラブル等に迅速に対応し、即交換させていただきます。

■表示価格は、税別表示です。詳しくは、お電話にて、お問い合せ下さい。掲載の価格は、10月下現在です。

ちゃんとパソコンしたい人の。

TSUKUMO

もう、離れられなくなるね。

あなたに、会いたい。

近日 **OPEN** 予定

うわさのパソコンロフト「ツクモパソコン本店」オープン

日本で初めての総合カメラ専門店「ツクモAV/カメラ館」はオープンセール中

PRESENT ツクモ全店で1万円以上お買い上げの方先着1万名様に越智静香チャンのフロッピーカレンダーをさしあげます。

静香チャンに会おう! 12月24日(月)PM1:00〜AV/カメラ館5Fのイベントフロアにて越智静香チャンのサイン会が開かれます。

ツクモパソコン本店
〒101 東京都千代田区外神田1-9-7 ☎03-253-5599

ツクモAV/カメラ館
〒101 東京都千代田区外神田1-11-3 ☎03-254-3999

パソコン本店OPEN！パソコン本店2FにX68000コーナーが移動してさらに充実！X68000ファンに応えます。

A Happy New Open!

掲載商品 2万円以上 送料無料!!

コンピュータでクリスマスカード&年賀状を ♡ ♡ ♡

SOFT

★CANVAS PRO-68K (CZ-249GS) 定価￥29,800
●CANVAS PRO-68K ドローグラフィックライブラリVol.1・2 ……定価各￥8,800
★NEW PrintShop PRO-68K (CZ-221HS) ……定価￥19,800

HARD 台/数/限/定

★24ピンドットプリンタ(80桁) Happy特価￥39,800 (消費税別途￥1,194)
★48ピン熱転写カラー漢字プリンタ Happy特価￥59,800 (消費税別途￥1,794)

X68000シリーズ

PROⅡ CZ-653C 定価￥285,000 / CZ-663C 定価￥395,000
EXPERTⅡ CZ-603C 定価￥338,000 / CZ-613C 定価￥448,000
SUPER HD CZ-623C 定価￥498,000

本店オープン記念特価にて提供中！是非おたずね下さい。

各種新製品 大特価販売中!!詳しくはお問い合せ下さい。

NEC・エプソン・東芝・富士通・各メーカー取り扱っています。

ハードディスク目玉品

40MB SASIタイプ Happy特価￥59,800 (消費税別途￥1,794)

光磁気ディスクユニット 台数限定

ソニー NWP-539N(光磁気ディスクユニット) ￥440,000
シャープ CZ-6BS1(SCSIボード) ……￥29,800
SCSIケーブル ……￥5,000
光磁気ディスク ……サービス(￥30,000)
合計定価￥509,800
Happy特価￥408,000 (消費税別途￥12,240)
クレジット例(48回払・税込)初回￥13,110+月々￥11,300×47回

ミュージックツールNEWプラス

お得なMIDIセット

Aセット
CM-32L ……￥69,000
SX-68M ……￥19,800
Musicstudio Mu-1 ￥19,800
合計定価￥108,600
Happy特価￥88,000 (消費税別途￥2,640)
クレジット例(18回払・税込) 初回￥7,223+月々￥5,600×17回

Bセット
CM-64 ……￥129,000
SX-68M ……￥19,800
Musicstudio Mu-1 ……￥19,800
合計定価￥168,600
Happy特価￥138,000 (消費税別途￥4,140)
クレジット例(24回払・税込) 初回￥7,603+月々￥6,900×23回

※「Musicstudio PRO-68K V1.1」又は「MUSIC PRO-68K(MIDI)」のソフトに変更の場合には￥8,000プラスになります。但し、これらのソフトがバージョンアップされた場合には変更する場合がございます。

追加オプション機器
●はなうたくんCP-40 ……定価￥33,000
●MIDIキーボード・コントローラーPC-200 定価￥36,000

アートツール

ハードウェア 限定品

一流メーカーイメージスキャナ 大特価目玉品 ￥128,000 (消費税別途￥3,840)
ビデオボード CZ-6BV1 定価￥21,000
カラーイメージユニット CZ-6VT1 定価￥69,800
数値演算プロセッサボード CZ-6BP1 定価￥79,800

ソフトウェア

Z's STAFF PRO-68K Ver2 Happy特価￥49,300 (消費税別途￥1,479)
マジックパレット Happy特価￥16,800 (消費税別途￥504)
サイクロンExpress α68 Happy特価￥83,300 (消費税別途￥2,499)
デジタルクラフト Happy特価￥38,800 (消費税別途￥1,164)

メモリーボード(X68000用)

(ACE & PROシリーズ用)
1MB増設RAMボード Happy特価￥19,800 (消費税別途￥594)
2MB増設RAMボード Happy特価￥39,800 (消費税別途￥1,194)
4MB増設RAMボード Happy特価￥69,800 (消費税別途￥2,094)

ハードディスク

★SASIタイプハードディスク
アイテック IT X640 特価￥84,800 (消費税別途￥2,544)
アイテック IT X880 特価￥99,800 (消費税別途￥2,994)
(カラー：ブラック・グレー)
★SCSIタイプハードディスク
アイテック IT X80S ……定価￥128,000 特価￥102,000 (消費税別途￥3,060)
アイテック IT X130S ……定価￥158,000 特価￥128,000 (消費税別途￥3,840)
(CZ-6BS1SCSIボードは別売 定価￥29,800)

光磁気ディスクユニット シャープCZ-6MO1 好評発売中！
SCSIボード シャープCZ-6BS1 好評発売中！

コミュニケーションツール

モデム 一流メーカー 2400ボークラス4 超・特・価
Happy特価￥28,000 (消費税別途￥840)

通信ソフトウェア
た〜みのる2 ……Happy特価￥15,000
Communication PRO-68K Ver.2.0 ……定価￥19,800

ツクモグローバルカード

大/好/評/入/会/者/募/集/！

国内・外で大活躍！
使って便利、持ってて安心／ツクモグローバルカードはジャックス・VISA、セントラル・マスターとの提携カードです。ツクモ各店でのお買物がらくらくできるうえに、国内はもとより海外でのショッピングもOK／

冬のボーナス一括払・金利手数料無！
受付中!! お申し込みは……(03)251-9898又は各店で！

18才以上なら学生でもOK!!

情報ツール

電子手帳 新・製・品
PA-9500 定価￥48,000 Happy特価販売中／
PA-8600 定価￥28,000 Happy特価￥24,800
接続ケーブル CE-300L 定価￥2,800 Happy特価￥2,500

電子手帳活用ソフト
CYBERNOTE PRO-68K 定価￥19,800
Stationary PRO-68K 定価￥14,800

開発ツール

C compiler PRO-68K Ver.2.0 好評発売中！ 定価￥44,800
SX-WINDOW 定価￥6,800

ビジネスツール

Hyper WORD 定価￥39,800
CARD PRO-68K 定価￥29,800
Kamikaze Happy特価￥57,800

ツクモ通信販売部 フリーダイヤル受注専用 **0120-377-999**

商品についてのお問い合せは各店店頭又は ☎03(251)9911へ

秋葉原各店 営AM10:15〜PM7:00

至お茶ノ水 昌平橋通り パソコン本店 5号店 7号店 万世橋 シントク電機さん 三菱銀行 中央通り ニューセンター店 AV/カメラ館 至神田 秋葉原駅 至上野 JR山手・京浜東北線 至浅草橋

★表示価格には消費税は含まれておりません。

ツクモは「スーパーX PRO SHOP」です。

PRO STAFF ツクモ

九十九電機㈱
〒101-91 東京都千代田区神田郵便局私書箱135号

パソコン本店 荒井
N・C店 福地

ツクモパソコン本店☎03-253-5599 (担当／荒井)

便利で安心な通信販売
通信販売部☎03-251-9911

■ツクモAV/カメラ館 ☎ 03-254-3999 (担当/川名)
■ツクモニューセンター店 ☎ 03-251-0987 (担当/福地)
■ツクモ5号店 ☎ 03-251-0531 (担当/森)
■名古屋1号店 ☎052-263-1655 (担当/吉高)
■名古屋2号店 ☎052-251-3399 (担当/横山)
■ツクモ札幌 ☎011-241-2299 (担当/村井)

★商品のご注文は在庫確認の上お願いします。

カード払い	全国代金引き換え配達	クレジット払い	現金書留払い	銀行振込払い	各種リース払い
通信販売での御利用カード、ツクモグローバルカード、VIPカード、セントラル、ジャックス※御本人様より電話で通信販売部へお申し込み下さい。	お申し込みは☎03-251-9911へお電話1本！配達日の指定もできます。	月々￥3,000以上の均等払いも頭金なし、夏・冬ボーナス2回払いも受付中！	〒101-91 東京都千代田区神田郵便局私書箱135号 ツクモ通販センター Oh!X係	事前に☎でお届け先をご連絡下さい。富士銀行 神田支店(普)No.894047 ツクモデンキ	くわしくは各店にお問い合せ下さい。ケースに合わせてご相談にのらせて頂きます。

超低金利クレジットOK.!

注目!!

《翌月一括払い(12月末)はもちろん》
冬のボーナス一括払い 手数料(金利)無料
(平成3年1月末払いをご利用下さい。)

またまた秋葉原でおなじみの

11/15~12/14

●お近くの方はお
●本体単品で特
●ビジネスソフト定

プリンター 10台限定 (送料¥1,000)
■CZ-8PK8 (定価¥152,000)
●24ピン漢字プリンタ (136桁)
●ハガキ印字OK!!
P&A 限定特価**¥49,800** (送料・消費税込¥52,324)

CYBER STICK
●CZ-8NJ2 (定価¥23,800)
超特価!!
¥18,500 (送料・消費税込み¥19,570)

X68000シリーズ専用 MIDIインターフェースボード
SX-68M (サコム)
(純生コンパチ)定価¥19,800
特価¥16,480
送料・消費税込み!

X68000用メモリーボード (I/O・DATA)(送料¥500)
①PIO-6BE1-A 定価¥25,000 (送料・消費税込¥19,055) **¥18,000**
②PIO-6BE2-2M 定価¥50,000 (送料・消費税込¥38,110) **¥36,500**
③PIO-6BE4-4M 定価¥88,000 (送料・消費税込¥66,744) **¥64,300**

ジョイスティック 送料¥500
●X-1PRO 定価¥9,500▶特価¥7,800
●ASCII STICK 定価¥6,800▶特価¥5,500

P&A超低金利クレジットをご利用ください!!

NEW X68000EXPERT II/II-HD & PRO II/PRO II-HD & SUPER-HD (送料・消費税込)

EXPERT II
セットでお買い上げの方に、
●ディスケット10枚
●ジョイカード2ケ
プレゼント中!!

EXPERT II

セット	構成	定価	12回	24回	36回	48回	60回
Ⓐセット	CZ-603C+CZ-604D	定価¥432,800▶特価(価格はお電話下さい。)	26,800	14,100	9,800	7,600	6,400
Ⓑセット	CZ-603C+CZ-605D	定価¥453,000▶特価(価格はお電話下さい。)	28,300	15,000	10,400	8,100	6,800
Ⓒセット	CZ-603C+CZ-613D	定価¥473,000▶特価(価格はお電話下さい。)	29,800	15,700	10,900	8,500	7,100
Ⓓセット	CZ-603C+CU-21HD	定価¥486,000▶特価(価格はお電話下さい。)	30,500	16,100	11,100	8,700	7,300

EXPERT II-HD
セットでお買い上げの方に、
●ディスケット10枚
●ジョイカード2ケ
プレゼント中!!

EXPERT II-HD

セット	構成	定価	12回	24回	36回	48回	60回
Ⓐセット	CZ-613C+CZ-604D	定価¥542,800▶特価(価格はお電話下さい。)	?	?	?	?	?
Ⓑセット	CZ-613C+CZ-605D	定価¥563,000▶特価(価格はお電話下さい。)	36,300	19,200	13,300	10,400	8,700
Ⓒセット	CZ-613C+CZ-613D	定価¥583,000▶特価(価格はお電話下さい。)	?	?	?	?	?
Ⓓセット	CZ-613C+CU-21HD	定価¥596,000▶特価(価格はお電話下さい。)	?	?	?	?	?

PRO II
セットでお買い上げの方に、
●ディスケット10枚
●ジョイカード2ケ
プレゼント中!!

PRO II

セット	構成	定価	12回	24回	36回	48回	60回
Ⓐセット	CZ-653C+CZ-604D	定価¥379,800▶特価(価格はお電話下さい。)	?	?	?	?	?
Ⓑセット	CZ-653C+CZ-605D	定価¥400,000▶特価(価格はお電話下さい。)	25,100	13,300	9,200	7,200	6,100
Ⓒセット	CZ-653C+CZ-613D	定価¥420,000▶特価(価格はお電話下さい。)	?	?	?	?	?
Ⓓセット	CZ-653C+CU-21HD	定価¥433,000▶特価(価格はお電話下さい。)	?	?	?	?	?

PRO II-HD
セットでお買い上げの方に、
●ディスケット10枚
●ジョイカード2ケ
プレゼント中!!

PRO II-HD

セット	構成	定価	12回	24回	36回	48回	60回
Ⓐセット	CZ-663C+CZ-604D	定価¥489,800▶特価(価格はお電話下さい。)	?	?	?	?	?
Ⓑセット	CZ-663C+CZ-605D	定価¥510,000▶特価(価格はお電話下さい。)	32,900	17,400	12,100	9,500	7,900
Ⓒセット	CZ-663C+CZ-613D	定価¥530,000▶特価(価格はお電話下さい。)	?	?	?	?	?
Ⓓセット	CZ-663C+CU-21HD	定価¥543,000▶特価(価格はお電話下さい。)	?	?	?	?	?

◎電話にて、ドンドンお問合せ下さい!!
クレジット表には、出せないほどの価格です。
メーカーさん、ご免なさい。
ユーザーの方には大歓迎されそうです。
今がチャンスです、ハイ。

SUPER-HD
セットでお買い上げの方に、
●ディスケット10枚
●ジョイカード2ケ
プレゼント中!!

SUPER-HD

セット	構成	定価	12回	24回	36回	48回	60回
Ⓐセット	CZ-623TN+CZ-604D	定価¥592,800▶特価(価格はお電話下さい。)	?	?	?	?	?
Ⓑセット	CZ-623TN+CZ-605D	定価¥613,000▶特価(価格はお電話下さい。)	?	?	?	?	?
Ⓒセット	CZ-623TN+CZ-613D	定価¥633,000▶特価(価格はお電話下さい。)	40,700	21,500	14,900	11,700	9,800
Ⓓセット	CZ-623TN+CU-21HD	定価¥646,000▶特価(価格はお電話下さい。)	?	?	?	?	?

X68000シリーズ ~P&Aスペシャルセット=限定誌上販売!!

EXPERT-HD
《今月の限定おすすめセット》

台数限定 送料、消費税込み
セットでお買上げの方に、●ディスケット10枚 ●ジョイカード2個 プレゼント中

Ⓐセット:P&A厳選セット
■CZ-612C(ブラック)
(本体価格¥466,000)
+
■CZ-604D(ブラック)
●モニター:スピーカー2個 チルト台付
(定価¥94,800)
P&A超特価 **¥340,000**

Ⓑセット:
■CZ-612+CZ-605D
定価¥581,000…▶特価**¥359,000**

Ⓒセット:
■CZ-612C+CZ-613D
定価¥601,000…▶特価**¥372,000**

Ⓓセット:
■CZ-612C+CU-21HD
定価¥614,000…▶特価**¥386,000**

■モデム 限定
⊙PV-A24VM5 (アイワ)
●MNPクラス5
定価¥44,800
特価**¥29,900**
(送料・消費税込¥31,827)

■40M・外付HDD
⊙WD-40 (ウィンテック)
●SASI
●ケーブル付 限定
定価¥102,000
特価**¥58,000**
(送料・消費税込¥60,770)

●本広告の掲載の商品の価格については、消費税は含まれておりません。●営業時間=平日AM10:00~PM7:00、日祭AM10:00~PM6:00

回～84回払いまでOK.!!

★頭金なし！★即日発送

●価格は流通事情により変動致しますので、銀行振込・書留等の送付前に、あらかじめお電話にてご確認下さい。

P&Aがズバリ超特価セールでご奉仕!!

立寄り下さい。専門係員が説明いたします。
価で受付します。詳しくは電話にてお問合せ下さい。
価の20%引きOK！ TELください。

全国通販

超特価でクレジットが組める!!

X68000用ソフトコーナー（送料1ヶ～5ヶまで￥500）

商品	定価	特価
Z's STAFF PRO68K Ver2.0（ツァイト）	￥58,000	￥39,500
Z's TRIPHONY デジタルクラフト（ツァイト）	￥39,800	￥27,500
テラッツォ（ハミングバード）	￥19,400	~~￥15,840~~
KAMIKAZE（サムシング・グッド）	￥68,800	￥45,500
C&Professional Pack（マイクロウェアジャパン）	￥58,000	￥43,000
Final Ver3.2（エーエスピー）	￥38,000	~~￥30,400~~
C-compiler PRO68K Ver.2 CZ-245L	￥44,800	￥34,500
CARD PRO68K CZ226BS	￥29,800	TEL下さい。
C compiler PRO68K CZ211LS	￥39,800	￥28,500
OS-9/X68000 CZ219SS	￥29,800	￥20,700
AI-68K CZ234LS	￥188,000	TEL下さい。
THE 福袋 V2.0 CZ224LS	￥9,900	￥7,400
SOUND PRO68K	￥15,800	￥11,300
MUSIC PRO68K CZ213MS	￥15,800	￥13,300
Sampling PRO68K CZ215MS	￥17,800	￥12,500
MUSIC-studio PRO68K 237MS	￥15,800	TEL下さい。
MUSIC-PRO68K〔MIDI〕247MS	￥28,800	￥20,500
New-print Shop 221HS	￥19,800	TEL下さい。
Communication 223CS	￥19,800	￥9,800
Communication Ver.2 CZ-257CS	￥19,800	￥15,500
C-TRACE68 Ver.3.0（キャスト）	￥98,000	￥69,800
サイクロンEXPRESSα68	￥98,000	TEL下さい。
G68K Ver2 PRO	￥22,000	~~￥17,600~~
THE FILE PROFESSOR（ロゴシステム）	￥28,000	~~￥22,400~~
Gツール（ザインソフト）	￥28,000	~~￥22,400~~
たーみのる2（SPS）	￥17,800	~~￥14,200~~
マジックパレット（ミュージカルプラン）	￥19,800	~~￥15,800~~
Hyper word CZ-251BS	￥39,800	~~￥21,800~~

●ゲームソフト20%OFF OK.!!（一部ソフト除く）

周辺機器コーナー（送料￥500）

商品	定価	特価
Ⓐ CZ-8NSI	￥188,000	￥145,000
Ⓑ CZ-6VTI	￥69,800	￥54,000
Ⓒ CZ-6TU	￥33,100	￥25,000
Ⓓ BF-68PRO	￥19,800	￥15,500
Ⓔ CZ-6BEI	￥35,000	￥26,500
Ⓕ CZ-6BEIA	￥38,000	￥28,600
Ⓖ CZ-6BE2	￥79,800	￥60,000
Ⓗ CZ-6BE4	￥138,000	￥107,000
Ⓘ CZ-6BFI	￥49,800	￥38,200
Ⓙ CZ-6BPI	￥79,800	￥61,000
Ⓚ CZ-6BMI	￥26,800	￥20,300
Ⓛ CZ-6EBI	￥88,000	￥67,500
Ⓜ AN-S100	￥36,600	￥28,500
Ⓝ CZ-6SDI	￥44,800	￥35,000
Ⓞ CZ-8PC3	￥65,800	P&A超特価 TEL下さい。
Ⓟ CZ-8PC4	￥99,800	
Ⓠ CZ-8PG1	￥130,000	
Ⓡ CZ-8PG2	￥160,000	
Ⓢ CZ-8PK10	￥97,800	
Ⓣ CZ-6PVI	￥198,000	￥153,000
Ⓤ IO-735X	￥248,000	￥190,000
Ⓥ CZ-8BSI	￥23,800	￥19,000
Ⓦ PIO-6BE1-A（I/O DATA）	￥25,000	￥18,000
Ⓧ PIO-6BE2-2M（I/O DATA）	￥50,000	￥36,500
Ⓨ PIO-6BE4-4M（I/O DATA）	￥88,000	￥64,300

X68000用ハードディスク（送料￥1,000）

アイテム
- ●HXD-040（40MB/23ms）……定価￥118,000▶特価￥ 88,000
- ●HXD-042（増設用）……定価￥128,000▶特価￥ 95,000

アイテック
- ●ITX-640（40MB/28ms）……定価￥158,000▶特価￥ 83,000
- ●ITX-680（80MB/20ms）……定価￥198,000▶特価￥103,000

プリンター（ケーブル・用紙付）限定5台 新品（送料￥1,000）

- ●CZ-8PC3（カラー漢字24ドット熱転写プリンター） 定価￥65,800……特価￥39,800
- ●CZ-8PK8（24ピン漢字プリンター136桁） 定価￥152,000……特価￥49,800
- ●CZ-8PC4 P&A特選!!（カラー漢字48ドット熱転写プリンター） 定価￥99,800……特価￥58,000

モデムコーナー （送料￥1,000）

商品	定価	特価
Ⓐ MD-24FS5（オムロン）	￥49,800	￥34,800
Ⓑ MD-24FS7（オムロン）	￥64,800	￥45,000
Ⓒ コムスター2424/4（NEC）	￥38,800	￥28,000
Ⓓ コムスター2424/5（NEC）	￥44,800	￥32,000

P & A 特選パソコンラック （送料無料）移動自由（キャスター付）

中古パソコン（セットはモニター付）送料￥2,000

- ●X68000セット……▶￥180,000
- ●X68000ACEセット……▶￥200,000
- ●X68000ACE-HDセット……▶￥215,000
- ● EXPERTセット……▶￥230,000
- ● EXPERT-HDセット……▶￥265,000
- ● PROセット……▶￥250,000
- ●X68000PRO-HDセット……▶￥270,000
- ● EXPERTⅡセット……▶￥250,000
- ● EXPERTⅡ-HDセット……▶￥320,000
- ● PROⅡセット……▶￥240,000
- ● PROⅡ-HDセット……▶￥310,000

中古パソコンはP&Aにおまかせ!!

その場で高価現金買取り・高価下取りOK.!!

■まずはお電話下さい。
03-651-1884 FAX:03-651-0141

■下取り・買取りでお急ぎの方、直接当社に来店、または、宅急便にてお送り下さい。

●下取りの場合……価格は常に変動していますので査定額をお電話で確認して下さい。（差額は、P&A超低金利クレジットをご利用下さい。）

●買取りの場合……現品が着き次第、2日以内に買取り金額を連絡し、振込み、又は書留でお送り致します。

●近郊の方は、P&A本店まで、直接お持ち下さい。即金にて、￥1,000,000までお支払い致します。

《便利な超低金利クレジットをご利用下さい》

- ●月々￥1,000円からOK.!!
- ●ボーナス払いOK（夏冬10回までOK）
- ●支払い回数 1回～84回
- ●お支払いは、8ヶ月先からでもOK.!!

通信販売お申し込みのご案内

〔現金一括でお申し込みの方〕
●商品名およびお客様の住所・氏名・電話番号をご記入の上、代金を当社まで、現金書留でお送りください。（プリンター・フロッピーの場合、本体使用機種名を明記のこと）

〔銀行振込でお申し込みの方〕
●銀行振込ご希望の方は必ずお振込みの前にお電話にてお客様のご住所・お名前・商品名等をお知らせください。
（電信扱いでお振込み下さい。）

〔振込先〕住友銀行 新小岩支店
当No.263914 ㈱ピー・アンド・エー

〔クレジットでお申し込みの方〕
●電話にてお申し込みください。クレジット申し込み用紙をお送りいたしますので、ご記入の上、当社までお送りください。
●現金特別価格でクレジットが利用できます。残金のみに金利がかかります。
●1回～84回払いまで出来ます。但し、1回のお支払い額は￥1000円以上。

超低金利クレジット率

回数	3	6	10	12	18	24	36	48	60	72	84
手数料	3.0	4.0	5.5	5.5	10.0	11.5	16.0	21.0	27.0	35.0	42.0

アフターサービス万全
全商品保証付。専門の担当者がお客様の立場で対応します。
初期不良、輸送トラブル etc.
万が一初期不良、輸送トラブルが発生しました際には、即交換させていただきます。

●定休日/毎週水曜日＝第3水曜・木曜は連休とさせていただきます（祭日の場合は翌日になります）

●マイコン
●ビデオ
●ビデオテープ

P&A 株式会社ピー・アンド・エー
〒124 東京都葛飾区新小岩2丁目1番地19号
☎03-651-0148（代） FAX. 03-651-0141

営業時間
平日：AM10：00～PM7：00
日祭：AM10：00～PM6：00

●現金書留及び銀行振込でお申し込みの方は、上記商品の料金に3%加算の上でお申し込み下さい。詳しくは、お電話でお問い合せ下さい。

■店頭にて、ゲームソフト25％OFF!!（税別）、超低金利オクトハッピークレジットをご利用下さい!!

■朗報です。冬のボーナス一括払い（12月末）OK!!手数料なし。（1月末払いもOK!!）ご利用下さい。

■便利です。夜9時まで営業しております。お立寄り下さい。お待ちしております!!

パソコンプラザ

案内図

至品川
東口ロータリー
JR蒲田から徒歩5分
京浜蒲田から徒歩1分
至品川
三菱BK
2F
商店街アーケード
JR蒲田駅
富士BK
三和BK
日本信託
八幡神社
京浜蒲田駅
環八通り
NTT
至横浜
至横浜

店頭セール実施中

'90オクトで始まるパソコンワールド

☎03-730-6271

●営業時間 AM 11:00～9:00／日曜・祭日PM7:00　電話一本で、ハイ即納

〒144 東京都大田区蒲田4-6-7 FAX 03-730-6273

●定休日毎週火曜日 祭日の場合翌日になります。

全国通販

オクト ラクラククレジット

1回	2.06	3回	3	6回	4	10回	5.5	12回	5.5	15回	8	18回	10
20回	11	24回	12	30回	16	36回	17	48回	22	60回	28		

OCT-1 システム インフォメーション

- ▶全商品保証付（メーカー保証）
- ▶超低金利ハッピークレジット（1回～60回）頭金ナシOK！
- ▶ボーナス一括払いOK！ボーナス2回払いOK.!!
- ▶配達日の指定OK！（万全なサポート体制）
- ▶商品の組合せ自由！ オクトフリーダムシステム
- ▶店頭デモンストレーション実施中

オクト セレクテッドシステム

広告掲載商品以外の製品も取扱っております。

蒲田

●冬のボーナス一括払い(12月末)OK!!

手数料なし!! 平成3年1月末払いもOK!!

▶今月のセットは、超お買徳!! 電話で交渉すべし!!

OPEN

★下記セットでお買い上げの方にはプレゼント!! ●①MD-2HD 10枚②ジョイカード 2個（連射式）③シリコンキーボードカバー ④ゲームソフト サンダーブレード（￥9500）

お好みのセットをお選び下さい。　送料無料

●SX-WINDOW搭載。
●40Mバイトハードディスク搭載

EXPERTⅡ・EXPERTⅡ-HD

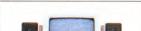
- ●CZ-603C-BK/GY 定価￥338,000
- ●CZ-613C-BK/GY 定価￥448,000

現金特価!! 推選 お電話下さい。

●SX-WINDOW搭載。
●拡張I/Oポート4スロット装備

PROⅡ・PROⅡ-HD

- ●CZ-653C-BK/GY 定価￥285,000
- ●CZ-663C-BK/GY 定価￥395,000

CZ-8NJ2 限定
●インテリジェントコントローラ
定価￥23,800
超特価￥18,600

15型カラーディスプレイTV

CZ-605D-GY/BK
定価￥115,000

15型カラーディスプレイTV

CZ-613D-GY/BK
定価￥135,000

14型カラーディスプレー

CZ-604D-GY/BK
定価￥94,8000

21型カラーディスプレイ

CU-21HD
定価￥148,000

セット	定価合計		12回	24回	36回	48回
Ⓐ CZ-603C＋CZ-605D	￥453,000	▶オクト大特価	?	?	?	?
Ⓑ CZ-613C＋CZ-605D	￥563,000	▶オクト大特価	?	?	?	?
Ⓒ CZ-653C＋CZ-605D	￥400,000	▶オクト大特価	?	?	?	?
Ⓓ CZ-663C＋CZ-605D	￥510,000	▶オクト大特価	?	?	?	?
Ⓔ CZ-603C＋CZ-613D	￥473,000	▶オクト大特価	?	?	?	?
Ⓕ CZ-613C＋CZ-613D	￥583,000	▶オクト大特価	?	?	?	?
Ⓖ CZ-653C＋CZ-613D	￥420,000	▶オクト大特価	?	?	?	?
Ⓗ CZ-663C＋CZ-613D	￥530,000	▶オクト大特価	?	?	?	?
Ⓘ CZ-603C＋CZ-604D	￥429,800	▶オクト大特価	￥28,000	￥14,800	￥10,200	￥8,000
Ⓙ CZ-613C＋CZ-604D	￥542,000	▶オクト大特価	￥36,000	￥19,000	￥13,100	￥10,200
Ⓚ CZ-653C＋CZ-604D	￥379,800	▶オクト大特価	￥25,400	￥13,400	￥9,300	￥7,200
Ⓛ CZ-663C＋CZ-604D	￥489,800	▶オクト大特価	￥32,200	￥17,000	￥11,800	￥9,200
Ⓜ CZ-603C＋CU-21HD	￥486,000	▶オクト大特価	?	?	?	?
Ⓝ CZ-613C＋CU-21HD	￥596,000	▶オクト大特価	?	?	?	?
Ⓞ CZ-653C＋CU-21HD	￥433,000	▶オクト大特価	?	?	?	?
Ⓟ CZ-663C＋CU-21HD	￥543,000	▶オクト大特価	?	?	?	?

♡本体セットは、1ヶ月間だけの大特価セール!! 11/15～12/14

♡クレジット価格は、消費税込みですヨ。ご利用下さい!!

※クレジットの回数は1回～60回、ボーナス併用などありますのでお電話でお問合せ下さい。

■本体セット：送料無料（注）本体セット以外の周辺機器（プリンター、モデム、HDD等）及びソフトの送料は、北海道・九州地区＝1ケロ￥1500、■その他離島地区は、1ケロ￥2000となります。

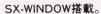
※上記料金には、消費税は含まれておりません。消費税が付加されますので、詳しくは、電話でお問合せ下さい。

■店頭にて、ゲームソフト25%OFF!!(税別)、超低金利 ハッピークレジットをご利用ください!!
■特に人気のある商品によっては、しばらくお待ち願うことがありますのでご了承下さい。

■平成2年冬のボーナス一括払いOK!!(12月/1月末)手数料ナシ!! 超低金利クレジットをご利用下さい。

厳選された製品を、より安く、より早く、皆様のお手元に!!

広告掲載商品以外の製品も取扱っております。

チャンス! X68000・SUPER-HD(チタン)=好評・発売中 どんどんTEL下さいネ。 送料ナシ!!

● ザ・ワークステーションと呼ぶにふさわしいスーパーな68000.!! 新登場!!
SUPER-HD。

※プレゼント! ①MD-2HD10枚 ②サンダーブレード(¥9,500) ③ジョイカード(連射式) ④シリコンキーボード(¥2,800)

X68000 SUPER-HD

● CZ-623C-TN+CZ-613D-TN
定価合計¥633,000…**大特価!! TEL下さい。**

※マウス・トラックボール付!! ディスプレイにはスピーカ2個、チルト台付!!

12回	?	24回	?	36回	?	48回	?

他のディスプレイ① CZ-602D、② 612D、③ CZ-603D、④ CU-21HDの組合せもこざいますのでお問い合せ下さい。

♡安くてゴメンなさい。今だけヨ!!

※超低金利クレジットご利用下さい。1回〜60回払い、頭金ナシ! ボーナス1回払い、ボーナス2回払いOK!

X68000 EXPERT-HD

オクト限定スペシャルセット

ラストチャンス!! 早い者勝ち!!

- CZ-612C(BK)(¥466,000)
- CZ-602D(BK)(¥99,800)
- MD-2HD 10枚
- ジョイカード(連射式×2個)
- シリコンキーボード・カバー

オクト超特価 ¥364,000(送料・消費税込み!!)

※ディスプレイ=①CZ-604D ②CZ-605D ③CZ-613D ④CU-21HD との組合せもございます。TEL下さい。

オクト特選 シャープ周辺機器 (送料¥1,000)

- CZ-6BE1 IBM増設RAMボード……(¥35,000)▶特価¥26,500
- CZ-6BE1B 1MB増設RAMボード………(¥28,000)▶特価¥21,000
- CZ-6BE2 2MB増設RAMボード……(¥79,800)▶特価¥60,500
- CZ-6BE4 4MB増設RAMボード……(¥138,000)▶特価¥104,800
- CZ-6BF1 増設用RS-232Cボード……(¥49,800)▶特価¥38,500
- CZ-6BG1 GP-IBボード……………(¥59,800)▶特価¥45,000
- CZ-6BM1 MIDIボード……………(¥26,800)▶特価¥20,500
- CZ-6BN1 スキャナ用ハラレルボード‥(¥29,800)▶特価¥22,800
- CZ-6BP1 数値演算フロセッサボード(¥79,800)▶特価¥60,500
- CZ-6BO1 ユニバーサルI/Oボード…(¥39,800)▶特価¥30,500
- CZ-6EB1/BK 拡張I/Oボックス…………(¥88,000)▶特価¥66,800
- CZ-6VT1/BK カラーイメージ・ユニット…(¥69,800)▶特価¥53,000
- CZ-6BL2 LANボード………………(¥298,000)▶大特価
- CZ-8NM2A マウス…………………(¥68,800)▶特価¥5,300
- CZ-8NT1 マウストラックボール‥(¥98,800)▶特価¥7,500
- CZ-8NS1 カラーイメージスキャナ……(¥188,000)▶大特価
- CZ-6BC1 FAXボード…………(¥79,800)▶特価¥60,500
- CZ-8TM2 モデムユニット………(¥49,800)▶特価¥38,000
- CZ-64H 増設ハードディスク…(¥120,000)▶大特価
- CZ-6TU GY/BK RGBシステムチューナー……(¥33,100)▶特価¥25,000
- BF-68PRO 高性能CRTフィルター……(¥19,800)▶特価¥15,500
- SX-68M(システムサコム) MIDIボード…………(¥19,800)▶特価¥15,000
- PIO-68BE1-A(I/O DATA) 1MB増設RAMボード ……(¥25,000)▶特価¥18,000
- PIO-6BE2-2M(I/O DATA) 2MB増設RAMボード ……(¥50,000)▶特価¥36,500
- PIO-6BE4-4M(I/O DATA) 3MB増設RAMボード ……(¥88,000)▶特価¥64,300
- CZ-6BV1 ビデオボード………(¥21,000)▶特価¥15,800

オクト面白グッズ

アイテック(送料¥1,000)

- IT-X640(¥158,000)…………特価¥103,000
- IT-X680(¥198,000)…………特価¥134,000

モデムコーナー(送料¥1,000)

- MD-1200AIII…………特価¥14,800
- MD-24FS4…………特価¥31,500
- MD-24FS5…………特価¥34,800
- MD-24FP4…………特価¥27,900
- MD-12FS…………特価¥15,000

熱転写カラー漢字プリンター (ケーブル用紙付) 送料¥1,000

CZ-8PK10
24ピン カラー漢字プリンター136桁

①CZ-8PK10(24ピン漢字プリンター136桁)
定価¥97,800……大特価!! TEL下さい。
②CZ-8PG1(24ピンカラー漢字プリンター80桁)
定価¥130,000……大特価!! TEL下さい。
③CZ-8PG2(24ピンカラー漢字プリンター136桁)
定価¥160,000……大特価!! TEL下さい。
④IO-735×(カラーイメージシェット)
定価¥248,000……大特価!! TEL下さい。

パソコンラック 推奨 送料無料

①五段キャスター付

5段キャスター付
キーボードが収納できるから、手元でマウス操作がラクラクできる棚板5段のマルチに活用できるディスク。ウーン、こいつはデキル!
1325(H)×640(W)×700(D)

特価¥16,000

②四段キャスター付

4段キャスター付
どんなパソコンにもフレキシブルに対応! 使い易いデスクです。
1245(H)×614(W)×600(D)

特価¥12,000

③三段キャスター付

3段キャスター付
場所を選ばない簡易で便利なディスクです。
限定
1175(H)×640(W)×600(D)

特価¥8,800

X68000ソフト大セール実施中※ゲームソフトオール25%off 送料¥500

〈グラフィック〉● Z's STAFF PRO68K Ver.2.0 (シャフト)定価¥58,000
オクト特価¥40,000

〈データーベース〉● KAMIKAZE (サムシンググッド)定価¥68,000
オクト特価¥46,000

〈グラフィック〉● C-TRACE68 (キャスト)定価¥68,000
オクト特価¥51,000

〈C言語〉● C & Professional Pack (マイクロウェアジャパン)定価¥58,000
オクト特価¥44,000

〈グラフィック〉● サイクロン エキスプレス 定価¥78,000
オクト特価¥58,000

〈グラフィック〉● デジタルクラフト 定価¥39,800
オクト特価¥28,000

〈ワープロ〉● ハイパーワード 定価¥39,800 CZ-251BS
オクト特価¥29,800

型名	商品	定価	特価
CZ-211LS	Ccompiler PRO-68K	¥39,800	¥28,800
CZ-212BS	BUSINESS PRO-68K	¥68,000	¥48,000
CZ-213MS	MUSIC PRO68K	¥18,800	¥13,500
CZ-214MS	SOUND PRO-68K	¥15,800	¥11,500
CZ-215MS	Sampling PRO-68K	¥17,800	¥12,800
CZ-219SS	OS-9/X68000	¥29,800	¥21,000
CZ-220BS	DATA PRO-68K	¥58,000	¥41,000
CZ-257CS	Print Shop PRO68K.V.2	¥19,800	¥14,300
CZ-223CS	Communication PRO-68K	¥19,800	¥14,300
CZ-224LS	THE 福袋 V2.0	¥9,900	¥7,500
CZ-226BS	CARD PRO-68K	¥29,800	¥21,300
CZ-241BS	システム手帳リフィル集	¥9,800	¥7,500
CZ-242BS	活用フォーム集	¥9,800	¥7,500
CZ-244SS	Homan 68K Ver.2.0	¥9,800	¥7,500
CZ-247MS	MUSIC PRO-68K〔MIDI〕	¥28,800	¥20,800
CZ-240BS	Stationery PRO-68K	¥14,800	¥11,500
CZ-243BS	CYBER NOTE PRO-68K	¥19,800	¥15,200
EW		¥38,000	¥29,800
G-68K		¥14,800	¥11,400
E-68		¥19,800	¥15,300

★オクト今月だけの新品限定販売(各1台限) (送料¥1,000)

- CZ-611C(BK)定価¥399,800…………大特価¥218,000
- CZ-652C(BK)定価¥298,000…………大特価¥188,000
- CZ-662C(BK)定価¥408,000…………大特価¥248,000
- CZ-601D(BK)定価¥119,800…………大特価¥68,000
- CZ-601D(GY)定価¥119,800…………大特価¥68,000
- CZ-612D(GY)定価¥119,800…………大特価¥74,000
- IO-735 定価¥248,000…………大特価¥158,000

店頭ゲームソフトオール25%off! ビジネスソフト 25%より特価中

★通信販売お申込みのご案内★ 〒144 東京都大田区蒲田4-6-7 TEL:03-730-6271

お申込みはお電話でお願いします。お客様の〈住所〉〈氏名〉〈電話番号〉及び〈商品名〉をお知らせ下さい。● 入金確認後ただちに商品をご送付いたします。

現金一括払い
銀行振込:お近くの銀行より(電信扱い)にてお振込み下さい。
現金書留:封筒の中に住所・氏名・商品名をご記入の上当社までお送り下さい。

クレジット
専用お申込用紙をお送り致します。ので、必要事項をご記入、ご捺印の上ご返送下さい。手続きは簡単です。

オクト ラクラク クレジット表

1回	2.06%	3回	3%	6回	4%	10回	5.5%	12回	5.5%
15回	8%	18回	10%	20回	11%	24回	12%	30回	16%
36回	17%	48回	22%	60回	28%				

振込先
富士銀行 久ヶ原支店 当No.1824
三菱銀行 蒲田支店 当No.0278691
株式会社 億人(オクト)

※掲載の価格は変動しますので、まずは、お電話にてご確認ください。
※上記料金には、消費税は含まれておりません。消費税が付加されますので、詳しくは電話でお問合せ下さい。
※銀行振込、または、現金書留でご注文の際には、あらかじめ電話でご確認の上、お申し込み下さい。

お待ちどうさま。いよいよ登場です。

エレクトロニクスショウ&データショウ'90

10月，東京晴海の国際見本市会場において2つのショウが開催された。2日から6日までがエレクトロニクスショウ'90，そして10月22日から25日までがデータショウ'90。この模様を写真を中心に紹介していきたい。

エレクトロニクスショウ'90

❶エレクトロニクスショウ会場風景
❷明るいところでもよく見えるシャープの高輝度プロジェクタ
❸パナソニックの高速応答液晶モジュール
❹ハイビジョンはほとんどのメーカーが36型を展示
❺カシオはアウトドアな雰囲気で液晶テレビをアピール
❻MC68340を含むMC68000ファミリが勢揃い
❼CD-Iエレショウ推進協議会のブース
❽エレショウのイメージキャラクタ？　のオブジェ
❾富士通は新型ASPをスーパーモナコGPでデモンストレーション
❿鈴木亜久里選手でおなじみ，DynaBookの東芝はＦ１を

データショウ'90

⓫データショウ会場風景
⓬あの MacII（カラー版）が318,000円！SE 相当も10万円台に！
⓭巨大なハイパー電子手帳
⓮いかにもプリンタのエプソンという感じ
⓯鮮やかな色を見せるTFTカラー液晶搭載のAX386LC
⓰富士ゼロックスのプリンタ兼コピー機，FAX にもなる

同人ソフトばかりがアマチュアプログラムじゃないぞ。1987年5月の開発開始から何度かの中断を経て，いまだ未完成。でも，自分の趣味で作るからここまでこだわれる。本当の“USER'S WORK”を紹介しよう。

今月はちょっと趣向を変えてX1turbo専用の「グラディウス」を紹介する。画面を見れば一目瞭然，あのコナミのグラディウスだ。驚かれる方もいるだろう。すでにX1版のグラディウスはコナミから発売されている。でも，作ってしまったものはしょうがない。作者は東京都の横内威至君(17)だ。たったひとりでX1turbo版グラディウスを作り上げた。

とにかく凄い！

最初は投稿作品として送られてきたものだが，残念ながら版権問題を抱え，掲載できる大きさでもない（ディスク2枚組）。しかし，その完成度たるや尋常ではない。できるだけ多く画面を載せたのでじっくり見てほしい。デジタル8色のみでここまでグラフィックを再現した力量は驚嘆に価する。止め絵だけでなく，アニメーションパターンもかなり本物に近い。爆発パターンまで細かい。諸々の事情によりプログラムの配布ができないのが非常に残念である。

凄いのはグラフィックばかりではない。いや，逆にこれだけのグラフィックなら動きが重そうだと考える人もいるかもしれない。X68000を除いて，これまでパソコン用として発売されていたグラディウスは，処理速度の関係からかオプションの数を制限していたくらいなのだから。

しかし，編集室に届いたテスト版では，お遊びといいながらもオプションを最大12個つけてみせる。これですべてのオプションからレーザー，ミサイルを発射してもちゃんとゲームできる速度である。さほど重くはならない。さらに背景にはちゃんと3重スクロールの星が瞬きながら流れている。自作の音源ドライバからはFM8声のBGMに，PSGの効果音だ。

動きその他も，これまでX1用に市販されてきたシューティングゲームのなかでも，はたしてこれ以上のゲームがあっただろうか？と思うくらい凄い。ゼビウス，サンダーフォース，スーパーレイドック，ルクソールなどよくできたシューティングゲームと比べてもまったく遜色ない。

ここに画面写真で挙げたものはまだ未完成なのだが，なにより凄いのは細部へのこだわりだ。

各面が終了すると背景はゆっくりとフェードアウトする。最終面をクリアすると敵要塞が縮小表示されていき，ビックバイパーが飛び出すところまで作ってある。さらにはX68000でも省略されたアーケード版と同じネームエントリーが用意されている。

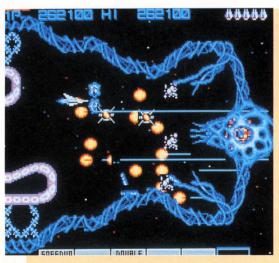
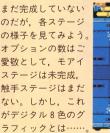
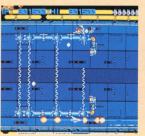
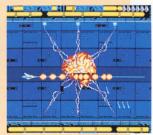
X68000版との比較という無謀な試みをして，あえて欠点を挙げると，

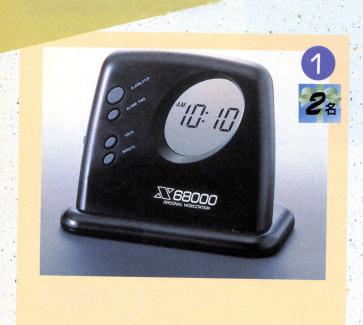
キャラクターがやや小さい
ザコキャラの動きが粗い
2，3面などで縦スクロールしない
スクロールが8ドット単位
一部のキャラで重ね合わせをしていない

といったところ。8ビット機だということとPCGの制約を考えればしかたないことではある。

X1turboでもこれだけのことができると証明した1作。X1turboにも不可能はないのか？

原作に忠実なグラフィック。この画面がしっかり動きます。本当に「グラフィックだけではありません」。本当に動きも凄いんだけどなあ……。

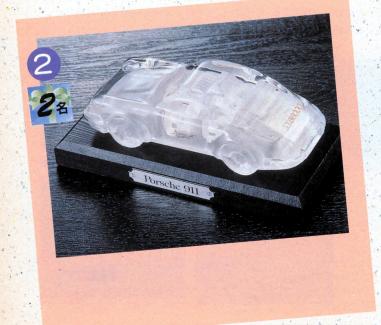
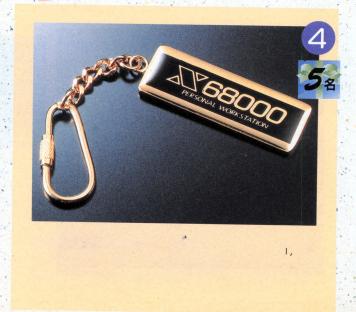
まだ完成していないのだが，各ステージの様子を見てみよう。オプションの数はご愛敬として，モアイステージは未完成，触手ステージはまだない。しかし，これがデジタル8色のグラフィックとは……。

Oh!X 3周年記念

愛読者プレゼント

X68000ジッポ・ライター

4,800円

ご存じX68000のイメージキャラクター（?），ツタンカーメンの絵入りオイルライター。持ったときの感触がなかかな

オリジナルソフト

5　5名

X68000ネクタイピン

3,000円

風に遊んでいるネクタイもいいけど，やっぱりオトコならひとつぐらいネクタイピンを持っていたい，そんなキミにあげたい品だ。

6　2名

X68000ポーチ

4,000円

ちょっと出掛けるときには便利なこのポーチ。外側にチャックもついていて，使いやすいことこのうえない。

7　2名

X68000ボストンバッグ

4,800円

旅行に出掛けるときにはこれ，ボストンバッグ。肩に掛けられるようにちゃんとベルトがついているのもうれしい。

8　1名

XBAS to C CHECKER

X68000用 5″2HD版

9,800円

X-BASICで書かれたプログラムをコンパイルしたときに出るエラーもこれで解消。これひとつでプログラムの修正もラクになるぞ。

C compiler PRO-68K ver2.0

9　1名

X68000用
5″2HD版

44,800円

バージョンアップしたC compiler PRO-68K。ソースデバッカをはじめとしてソフト開発には欠かせない機能を満載したソフトだ。

CANVAS PRO-68K

10　1名

X68000用 5″2HD版

29,800円

ドロー系グラフィックツールとして注目のこのツール。細かな曲線や，拡大縮小機能，変形もできる。印刷の美しさも見逃せない。

熱血高校ドッジボール部 サッカー編

11　3名

X68000用
5″2HD版2枚組

8,800円

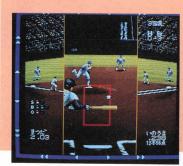
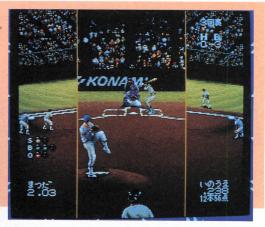
もうお馴染みのくにおくんシリーズ。今回は助っ人でサッカーに挑戦。もちろんサッカーのルールなんて知らなくても楽しめる。

画面は開発中のものです

パロディウスだ！
お馴染みグラディウスのパロディ版，パロディウスだ！　の登場。デフォルメされた動物たちがなんとも可愛い。

生中継68
コナミのX68000オリジナル野球ゲーム。野球中継風に投げるとき，打つときの画面が見られるのも面白い。

SOFTWARE INFORMATION

今年も年末年始にかけて発売されるゲームがもりだくさん。移植ものあり，オリジナルものあり，“2”ものありと，バラエティも豊かです。これで冬休みは退屈しないですむかな？

話題のソフトウェア

ふ，ふぁっくしょん，っと一，あ一，カゼひいたぁ。あ，いきなりドーモすいません，皆さんはお元気ですか？　突然寒くなったもんで，デリケートなあたしは気候の変化についていけなかったの。やっぱり美人薄命なのね……，くしゅんっ。んなカゼくらいでオーバーなって？　いいじゃない，一度いってみたかったんだから，このセリフ。

いやー，今月もゲームが多いこと多いこと。もう各ソフトハウスさんの年末にかけて意気込みが感じられますね。それじゃ，さっそくいってみましょー！

今月トップを飾るのはコナミ。2本一挙に発表です。まずは皆さんの期待に応えるべく登場の**パロディウスだ！**　今年アーケードで人気だったこのシューティングゲームが，いよいよX68000に移植です。プレイヤーは，おちゃめで可愛いマイキャラ（ビックバイパー，たこ，ツインビー，ペン太郎）を操り，これまた可愛い敵キャラを倒していくというもの。もちろん，この

やっぱりズームは強かったのだ

順位	タイトル	前回
1	ラグーン	2↑
2	シムシティー	1↓
3	サイバリオン	10↑
4	ソーサリアン（含追加シナリオ）	7↑
5	ポピュラス（含プロミストランド）	4↓
6	ダンジョン・マスター	－
7	エアー・コンバット〜遊撃王II〜	－初
8	パロディウスだ！	－初
9	機甲師団	－初
10	ワンダラーズ・フロム・イース	5↓

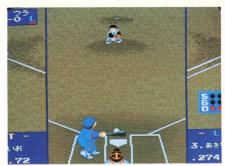
ラグーンがシムシティーを押さえてトップに躍り出ました。どちらも発売前から票を集めていただけに発売後の動向に興味津々でしたが，X68000ユーザーはやはりアクション要素のあるものが好きなようです。

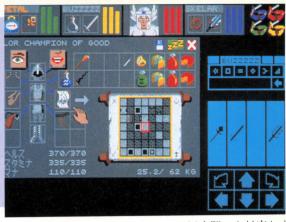
しかしラグーンは，支持が「すべていい！」という熱狂的な人と，「次回への期待票の意味を込めて」というやや冷めた目で見ている人に分かれています。どちらもズームのセンスには高い評価を下しているようですが，トップを維持できるかどうかはまだまだ流動的。

10位からジャンプアップして3位につけたのがサイバリオン。原作の独自性，移植の完成度について言及しているハガキが多いのですが，なかにはMIDIに対応していないのが不満という声も。そろそろMIDIも必須アイテムになりつつあるということでしょうか。

6位にはダンジョンマスターがカムバック。先月ポピュラスが4位でがんばっているのにランク外になったのがファンにとっては，心外だったらしく，「まだまだ遊んでます」というハガキがどっさりきました。

ワールドスタジアム

続ダンジョン・マスター カオスの逆襲
あのダンジョン・マスターの続編。今回はさらに入り組んだダンジョンが君を待つ。エディタで自分の好きな顔を描けるのもいい。

ゲームを知らない人でも察しがつくとおり，これはグラディウスのパロディ版。でも，そんなこと関係ナシにこの楽しい雰囲気を味わってほしい，そんな作品なのです。発売までお楽しみに。

で，もう１本はX68000オリジナル野球ゲーム**生中継68**。アニメーションやカメラアングルで，あたかもTVの野球中継をそのまま見ているような気にさせてくれるゲームです。エディット機能で細かいところまで設定変更でき，自分好みのチームを作れるのもうれしいところ。こちらは１月発売予定。もうちょっと待っててね。

続けてもう１本野球ゲームを紹介しちゃいましょう。こちらはナムコのアーケード版からSPSが移植した，**ワールドスタジアム**です。これはゲーセンで遊んだ人もかなりいるんじゃないでしょうか。移植にあたっては，音声データや効果音はアーケードとほぼ同じ，もちろんオールスター戦でのエディット機能も健在ということなので，期待していてもよさそう。これは要２Mバイトです。12月14日発売の予定。

12月14日発売予定といえば，こちらもそうビクター音楽産業の**続ダンジョン・マスター　カオスの逆襲**。これはタイトルどおり倒したはずのロード・カオスが，なんと生きていた，かくして勇者は再びダンジョンへ……という筋書き。もちろん「ダンジョン・マスター」で大切に育てたキミの勇者をこの続編でも使用できる。ただ，かなり強くなっているという設定上，あまり育っていない勇者だとそうとう苦しいはず。というわけで，もちろん新しい勇者もちゃんと用意されています。今回はダンジョンも難しいためオートマッピング機能もついています。そして新しい魔法や，デモアニメーション，それにFM音源にも対応しました。勇者の顔もエディタで描き変えられるので，一層親しみがわきますね。

さて，ピンボール・ピンボールを発売したばかりの日本ソフテックでは，なんともう第２弾を発売しちゃいます。早いですねー。今回もピンボールもので，タイトルは**LUCY・SHOT**です。前回の中国風とはうってかわって，今回はアメリカナイズされたグラフィックが印象的。

そして，ハミングバードの**ラプラスの魔**，これもやっと姿を現してきたよう。ゲームの内容はホラーRPG，画面写真からオドロオドロしい雰囲気が伝わったでしょうか。もう少ししたら詳しく紹介できそうなので，待っていてくださいね。

ブラザー工業（タケル）からは**ハイドライド３SV**（スペシャルバージョン）がすでに発売されています。皆さんもうプレイしてみましたか？　あのARPGの名作が

LUCY・SHOT

ラプラスの魔

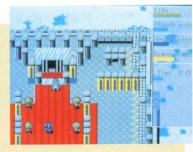
ハイドライド3SV

７～９位はすべて初登場。ハガキの声は……。

エアー・コンバット～遊撃王II～：サイバースティックを使うと気分が出る。操作がシンプルでなじみやすい。空中戦が面白い。ただスピード感はいまいち。

パロディウスだ！：写真で見ただけだが完成度が高いと思った。アーケードの原作が好きなので。グラディウスIIIも期待している。

機甲師団：きれいなグラフィックと難しめのシナリオがいい。もう少し速ければもっとよかった。ウォーシミュレーションとしては異色。

ところでTOP10では「キライだから」というハガキも１票に加えているので，本当にこのソフトはダメだと言いたいときは「編集部へのメッセージ」欄にお願いします。ちょっと不満も言っておきたいが好きだというのは「推薦する市販ソフト」欄でかまいません。では。（浦）

ブルトン・レイ

銀河英雄伝説II

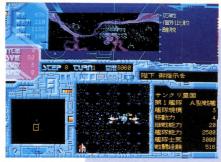
遥かなるオーガスタ
ご存じT&E SOFTの3Dゴルフゲーム。やっとX68000版の登場です。もちろん，マウスで簡単に操作できます。デキは上々，発売が待たれる一作です。

シュヴァルツシルト

X68000版として再び登場。こちらは来月号で詳しく紹介する予定ですので，お楽しみに。

RPGといえばこちらもそう，システムソフトの**ブルトン・レイ**。このゲームは中世のヨーロッパを舞台にしたRPGです。ゲームはなんと空の上から始まるというのだからなんともファンタジック。どんな世界が待っているかワクワクしちゃいます。

ガラっと変わってこちらはシミュレーション，ボーステックの**銀河英雄伝説II**です。こちらも発売間近，一層気合いが入っているようです。コマンドなどは前作とほぼ同様なので，悩まずにできそう。それに，指令官のデータは顔写真つきで紹介されるので，アニメからファンの人はより身近に感じられていいですよ。

シミュレーションといえばこれです，お待たせしました，PC-9801に遅れること約1年半，やっとT&E SOFTの**遥かなるオーガスタ**の登場です。T&E独自のポリシスという3Dシステムを使って開発されたこのゴルフゲーム，出来はやはりさすが，のひと言。これなら十分自信をもってオススメできます。グリーンの起伏をワイヤーで表したり，キャディさんのアドバイスなんかもあったりして親切設計されています。お父さんに「これって面白いんだよぉ」とかいって買ってもらう輩が増えそうですけどね。発売日はまだ未定，でもかなり仕上がっているので，おこづかいを貯めて待つことにしましょう。

さて，先月も紹介した工画堂スタジオの**シュヴァルツシルト**ですが，もうすぐ完成の予定。宇宙を舞台にしたウォーシミュレーションですが，ひとつ，またひとつと次々に目的が出てくるところが，ファンを引き込む要素となっているよう。年内には発売される予定。本誌でも，いのいちばんにレビューで取り上げるつもりなので，楽しみにしていてください。

そして，ジェミニウイングが好評のシステムサコムでは，もう次の作品**アトミック・ロボキッド**を開発中。こちらもアーケードからの移植で，UPLの作品です。デフォルメされたロボットを操りガンガン撃ちまくるシューティングですが，これがなかな

発売中のソフト

★NAIOUS

X68000にオリジナルのシューティングゲームが登場。エグザクトという新参ソフトハウスのデビュー作だ。スタイルはサイドビュータイプだが，場面ごとに縦にも斜めにもスクロールする。さらに，ラスタースクロールなどのテクニックも使いまくって，非常に演出に力を入れたものになっている。オプション4種類にパワーアップ5種類とプレイバリエーションも広く，デカキャラの迫力もなかなかで，期待が持てそうな1本である。詳しくは，レビューのほうでどうぞ。

X68000用　5″2HD版2枚組　8,700円(税別)
エグザクト　☎025(247)9160

★LUCY・SHOT

先頃発売されたばかりのピンボール・ピンボールの別バージョンがはやばやと登場。前回の渋い色調からはうってかわって，今度はアメコミ風の派手なグラフィックが背景を彩っている。もちろんパーツの構成も一新されているが，特定のホールに玉が入ると面が変わるというソフテックらしい仕掛けは従来どおり。全部で4面が用意されている。ピンボールはド派手でなきゃ，という人におすすめ。

X68000用　5″2HD版　7,800円(税別)
日本ソフテック　☎0425(82)1502

★3D倶楽部

3次元イメージシミュレータ「Z'sTRIFFONY・DIGITALCRAFT」用のデータライブラリが発売される。家具のデータが多数用意され，デジタルクラフトを有効に使うためのサンプルとしてだけではなく，部屋などの空間デザインに実際に活用することができる。

「ダイニングキッチン編」「リビング編」が10月中旬に登場し，追って「地図編」と「プライベートルーム編」も11月中旬に発売になる予定。

X68000用　5″2HD版　9,700円(税別)
ツァイト　☎03(299)0461

新作情報

★ダイナマイトデューク

レナム以来ごぶさたしていたヘルツが，アーケードからの移植をひっさげてきた。それがこの「ダイナマイトデューク」。シューティングと格闘をミックスしたアクションゲームだ。

プレイヤーは機械の右腕を持つ男，ダイナマイトデュークとなって生物兵器軍団の野望を阻止せんと敵アジトに潜入する。マシンガンを乱射し，必殺ダイナマイトパンチを炸裂させろ！　ボスキャラとの格闘シーンも用意され，なかなか激しいゲームとなっている。

X68000用　5″2HD版2枚組　価格未定
ヘルツ　☎03(371)3012

★パロディウスだ！

X68000ユーザーには馴染みが深い「グラディウス」シリーズのパロディ版，「パロディウスだ！」がコナミから発売される。これは，今年アーケードゲームとして登場した作品の移植版。スタイルはほぼ「グラディウス」シリーズの流れを継いでいるが，登場キャラクターが，ペンギンやネコから相撲取り，踊り子までちゃめっ気たっぷりのものに変更されている。ベルを使ったパワーアップなども組み込まれて，従来にない雰囲気を持つシューティングゲームとなっている。

X68000用　5″2HD版　価格未定
コナミ　☎03(262)9110

★遥かなるオーガスタ

本格的3Dゴルフシミュレーションゲームが年末に登場する。T&E SOFTが開発した「ポリシス」システムにより，コースの微妙な高低を表現しながら緻密な風景も同時に描くことに成功した。これにより，コースのどこからでも美しい3D画面を描けるという特徴を持っている。コースは世界4大トーナメントに数えられるマスターズの舞台，オーガスタナショナルゴルフクラブ。ストロークプレイのほか，マッチプレイやトーナメントでも遊ぶことができる。

X68000用　5″2HD版　12,800円(税別)
T&E SOFT　☎052(773)7770

★シュヴァルツシルト

宇宙を舞台に国家の興亡を描くSFシミュレーション。プレイヤーは多くの国家の中のひとつ，サンクリ星国の皇太子として銀河の統一を目指す。国内を発展させ，外交政策を駆使し，艦隊の増強をはかる。戦闘には要塞戦，艦隊戦，惑星戦などさまざまな種類があり，それらがアニメーションによって表現される。さまざまなハプニングを克

アトミック・ロボキッド

かタイヘン。またまたシューティングファンには見逃せない一作となりそうです。12月発売の予定です。

さて，今月3つ目の野球ゲームです。アートディンクでは，**栄冠は君に**を開発中。このゲームは先ほどの2つとは違い，高校野球を題材にしたもの。約4,000校の出場校を相手に，トーナメントを勝ち抜いていこうというもの。ユーザー優勝プレゼントに，深紅の大優勝旗も用意されているぞ。

さて，テーブルトークの老舗，ホビージャパンからは**リングマスター**IIの登場です。前作から1年たったいまでも固定ファンのいるリングマスター，今回もシステム的にはそう変わりなく，ファンを喜ばせてくれそう。まだ開発中ですが，画面を見ていただければ出来はわかっていただけると思います。こちらももうちょっとしたら詳しくお伝えするつもりです。

最後に画面は載せられなかったけど，タイトルだけでも紹介していきましょう。

まずはシムシティー以降の動きが気になるイマジニア，こちらは**シムアース**の開発に取り組んでいる様子。発売は来年3月あたりになりそう。

データウエストからは，お馴染みの「Misty」シリーズ，**Misty Vol.6**が発売されます。X68000，X1turbo同時発売で11月22日発売です。なお，タケルでも発売されます。

X68000オリジナルで話題のウルフ・チームでは，シューティングゲーム**ソル・フィース**を発売予定。詳しくは来月。

スタークラフトではPC-9801などで発売されている**ファンタジー**IVをX68000に移植開発中。発売は来年1～2月頃になりそう。

ゲーム以外にも，ツァイトからZ'sTRIPHONY・DIGITAL CRAFT のデータ集**3D倶楽部**が発売になりました。これはお部屋のインテリアパーツをいろいろと用意し，画面上に自分なりの部屋をデザインしようというもの。なかなか本格的ですよ。

そうそう，**C-TRACE68＋**もキャストから12月に発売されす。こちらはC-TRACEのバージョンアップ版。198,000円とちょっと高めだけれど，ついにメタボールが使えるぞ。

サン・ミュージカルサービスの**Mu-1**がバージョンアップ。FM音源関係や和音の確認機能など。登録ユーザーには無償バージョンアップされる模様。Musicstudioもバージョンアップされる予定。

それでは，また来月。

栄冠は君に

リングマスターII

服して，星系を我が物にするのだ！

X68000用　5"2HD版2枚組　12,800円(税別)
工画堂スタジオ　☎03(353)7724

★リングマスターII　永遠なる想い

前作「フィリアス・ノギスの暗雲」の主人公は，リングナイツの座を手に入れて再び旅立った。戦乱を予感した彼は，隣国ムルソン大公国へと向かう。が，そこでも新たな暗雲がたちこめていた……。充実した世界設定とシナリオ，綿密な戦闘システムを持つRPG。人物との会話に重点を置いてあり，テーブルトークの老舗らしい独特の雰囲気を持つゲームである。

X68000用　5"2HD版　価格未定
ホビージャパン　☎03(354)9341

★生中継68

コナミから久々のアクション型野球ゲームが登場。「生中継」というだけあって，X68000のグラフィック機能を活用して，試合の興奮をそのまま伝えることに主眼を置いている。画面のアングルはTV中継と同様にし，アニメーションによって臨場感を出す。エディット機能とゲームモードの種類も充実させて，ユーザーの細かい要求にも応えてくれそうだ。

X68000用　5"2HD版　価格未定
コナミ　☎03(262)9110

★ブルトン・レイ

中世ファンタジーの世界に基づいたフィールドタイプのRPG。登場人物すべてが自分の思考を持っている点が特徴で，戦闘もプレイヤー以外のキャラクターは自分で行動を決定する。ゲームは複数の短編シナリオからなり，レベルアップしたキャラクターやアイテムを，ほかのシナリオに引き継ぐこともできる。色調を抑えたグラフィックや，リュートを意識したBGMなど，中世ヨーロッパの雰囲気を出すことにも気を使った作品だ。

X68000用　5"2HD版2枚組　価格未定
システムソフト　☎092(752)2602

★ラプラスの魔

ラプラスの魔は，原作は安田均氏，音楽に小坂明子氏と豪華なスタッフを迎えて贈るホラーRPG。アメリカはマサチューセッツ州東部の「ウェザートップの館」という洋館の中で，ゴーストハンターと呼ばれる人々と怪物たちの戦いが繰り広げられる。そして，舞台は館を越えてさらに……。精神的なダメージなども考慮され，ホラーRPGらしいデザインに仕上げられている。

X68000用　5"2HD版2枚組　価格未定
ハミングバードソフト　☎06(315)0541

★栄冠は君に

アートディンクの新作は高校野球のシミュレーションゲーム。去年地区予選に参加した全国の高校約4,000校がすべて登場する。プレイヤーは監督となってチームを率い，部員の能力を把握しながら40日間の練習をこなし，地区予選，そして全国大会へと進んでいく。自信や気力などの高校野球らしい要素を加えて監督としての難しさを体験できるゲームだ。それだけに優勝したときに見るデモは感激ひとしお。

X68000用　5"2HD版3枚組　9,500円(税別)
アートディンク　☎0474(77)7541

★アトミック・ロボキッド

UPLのアーケードゲームがシステムサコムによってX68000に移植・発売される。左右スクロール型のアクションゲームだ。プレイヤー操るロボキッドは，人類の存亡の鍵を握るプログラムを持ちながら，その使命を知らないというなかなかハードな設定。全25面で，1画面分以上もあるボスキャラや，同型機との対戦もある。音楽はMIDIにも対応。

X68000用　5"2HD版　8,800円(税別)
システムサコム　☎03(635)7609

★ワールドスタジアム

ファミコン，アーケード，PCエンジンなどで好評を博したナムコの野球ゲーム，「ワールドスタジアム」がX68000にも登場する。12のプロ野球のパロディ球団，3つのスタジアムを選んで対戦する。ヒッティング，バント，牽制球やダブルスチールまで多彩なアクションを使い分けて勝利を目指そう。2人プレイも可能。要2Mバイト。

X68000用　5"2HD版　8,800円(税別)
SPS　☎0245(45)5777

★C-TRACE68＋

キャストのレイトレーシングソフトウェア「C-TRACE」がバージョンアップした。メタボールにより有機的質感が表現でき，αチャンネルへの対応により高度な合成作業を可能とした。光の当て方の設定も自由になり，照射範囲の境界をぼかすこともできる。

X68000用　5"2HD版　198,000円(税別)
キャスト　☎03(705)1065

シューティングゲームの王道である

Nakano Shuichi
中野 修一

あの「R-TYPE」のアイレムが，久々にX68000のゲームを発表。それもアーケードで秀作との呼び声の高い「イメージファイト」だ。もうすぐ発売，それまでウデを鍛えておこうではないか。

少し飢えていた，と思う。バリバリのガシガシのシューティングゲームに，だ。硬いボスキャラ，貧弱な武器，複雑な操作，なにがゲームバランスだ，アクションRPGだろうが！　えーい，一方的に撃ってくるんじゃねい！　どんどんフラストレーションがたまる。

ということで，イメージファイトだ。この年末にはいくつかシューティングゲームが発売されるが，ガンガン撃ちまくりたいという人にはイチ押しの作品。こと，こういうことをやらせるとX68000のハードウェア構成は強力だということを思いしらせてくれるぞ。うーん，2重スクロールにスプライトはあくまで滑らか。CYNTHIA(X68000のスプライトコントローラ）はいい仕事をしている。

イメージファイトとは

もともとイメージファイトとはアイレムが1988年に発表したシューティングゲーム。あのゲーム史に残る名作を作ったR-TYPE開発部隊の作品とあって注目を浴びた。それがアイレム自身の移植によってこのたびX68000にお目見えとなった。とにかくX68000では久々の縦スクロールシューティングゲームだ。めでたい。

まずプレイヤーはシミュレータでテストを繰り返し，それにパスしなければ実戦には参加できない（……こともないが，補習を受けなければならない)。前半の5ステージがシミュレータで後半の3ステージが実戦だ。要するに1～5がイメージで残りがファイトなわけだ(?)。

特徴的な攻撃方法を解説しよう。

まず，バラ弾を撃つ。ノーマルショットというやつだ。連射がきき，かなり強力。また，自機の速度を変えるとバックファイアーが発生し，後方の敵にダメージを与えることができる。これものちのち便利になることもある。

そしてオプション兵器。ポッドというものが自機の左右と後方，最大3機装着できる。青いポッドは前方に弾を撃ち，赤いポッドは進行方向の逆に弾を撃つ。赤いポッドとお友達になることがイメージファイトの醍醐味に通じる。狭い通路でもひょいと左右を撃ち分けたり，どてっ腹に弾を集中したり，クルクルとまわりじゅうに弾をばらまいたり，とにかく頼りになる。間違えて青いポッドを取ってしまったら，左右は体当たりに限る。ポッドシュートといって左右のポッドを前方に飛ばす技もある。

頭部には特殊兵器のユニットが装着できる。派手なリングレーザー，サーチレーザー，誘導ミサイル，ワイドレーザーなど好みで使い分ければよい。これらは1回分のバリアとしても役に立つ。

が，とりあえずポッドさえあればなんとかなる。飛び道具満載のシューティングゲームでも，ゲーム感覚は「近接戦闘」なのが面白いところ。

原作との違い

最初に水を差すようで悪いが，もともとアーケードというのは，とんでもない世界ではある。オーバースペックなハードウェアによってたかってソフトを載せる。パソコンで完全な移植なんてそうそうできるものではない。やはり原作と並べれば見劣りせざるをえない。そこんとこを踏まえたうえで読んでほしい。

アーケード版では縦画面だったこのゲーム，X68000では横画面に変更されている。よって，一部の場面でアーケード版のゲームバランスが再現されていない。が，ドラゴンスピリットの横画面が気にならない人ならまず問題ない。一部のボス部分だけはバックスクロールで対応しているようだ。

まだボスキャラではありません

画面はまったくゲーセン版に遜色ない。動きもやや粗いがまあ合格。高速スクロールがときどきガタつくことがある以外，全体の動きはかなり軽い。最初のうちは敵キャラの出現頻度などでもの足りなく思えることもあると思う。やがてそれは相当手加減されていたということに気づくだろう。極端にキャラが増えると途端に重くなるゲームが多いなかで，これだけ余裕を感じられるゲームも珍しい。R-TYPEとはえらい違いだ。

細かい部分で敵の硬さやバランスが変更されているようだが，厳として「イメージファイト」以外の何者でもないと感じさせる仕上がりである。PCエンジン版やファミコン版の忠実度とは比較にならない。

難易度は？

これだけ気合いの入った移植だからゲームも本格的だ。もともとゲーセンの達人ゲ

これがポッドシュートだ！

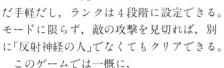
X68000用　5"2HD版2枚組　8,800円(税別)
アイレム　☎06(535)4880

ーマー用に作られたもの。そのままでは素人には手が出ない。そこで自動連射モード。これでハンデはなくなった。

ファミコン版，PCエンジン版でのHARDモードよりX68000版のEASYモードのほうが難度が高いという噂もあるが（事実だろう），1周目に限定すれば適正な難易度といえる。ドラゴンスピリットやサンダーフォースIIに比べればまだ手軽だし，ランクは4段階に設定できる。モードに限らず，敵の攻撃を見切れば，別に「反射神経の人」でなくてもクリアできる。

紙一重の美学。安全地帯がなんでい！

いよいよ大詰め。最終ステージへ

このゲームでは一概に，

難易度を上げる＝敵が硬くなる

ではなく，ランクを上げても個々の敵の硬さはあまり変わらない。変わるのは敵の出現頻度（と弾の速度）だ。とりあえず進みたいという人はEASYを，たくさん敵が出るほうが得した気になる人はランクを上げたほうがいい。実はVERY HARDでも最高に難しいわけではない。どうせ，EASYでも2周目になれば1周目のVERY HARD以上の難易度になるのだ。

また，ノーコンティニューかどうかでキャラクターの出方も変わる。途中で中ボスひとつ加わるだけでもゲームの緊迫感が違ってくる。さらに，赤いポッドを取ると難易度が上がるという噂もある（未確認）。

ちなみに，アイレムの伝統かどうかは知らないが3周目はない。2周目をクリアした時点でゲームは終了する。まあ，無限コンティニュー制とはいえ，一般庶民にはあまり縁がない世界ではある。

また，イメージファイトの特徴としてシミュレータでの撃墜率が90%以下なら最終面より難しいといわれる補習ステージをこなさなければならないことがある。

つらい。まずパターンを見切って攻略法を摸索する。と，できあがったプログラムには一瞬の遅れも許されないということがわかる。敵の出現予定位置に先回りし，密着して連射，敵が爆発を始めるタイミングで次のターゲットに移動。場合によっては，一瞬だけ飛び出して敵の攻撃を誘導しておかなければかわせないこともある。もちろん，スピードは3以上だ。複合攻撃をいかに避けるかがポイントになりそう。

各面をきちんとクリアすれば縁がないが，一度は挑戦してみたいステージだ。

補習ステージの光景

ここがいい！

現れるものをすべて破壊しなければならない。逆にいえば，すべて破壊できる，というコンセプトがいい。敵の攻撃ばかり激しくて，撃ちまくるより逃げまくることが要求されるゲームが多い昨今，こういうストレートな作りが新鮮だ。

R-TYPEにもいえたことだが，フル装備状態で死んでも途端に絶望的になったりはしない。どこからでもちゃんと復活できるのがいい。復活できる。信じればできる。

敵キャラはザコでも大型で，なおかつアニメーションパターンも多い。なによりデザイン，描き込みが美しい。とにかく派手で綺麗なのがいい。

キャラクターのレスポンスがいい。無骨なデザインの自機もいい。左右移動で機体がアニメーションするのは当然として，速度を変えれば「ばふばふ」と変形し，移動方向によってポッドも微妙に位置を変える。なにか操作するとそれが絵になって現れる。描き込みは凄いけど，どこか平面的なゲームというのもあるが，動きがいいと自然と奥行きが感じられるものだ。

＊　＊　＊

セリーグの野球と大リーグを比べるようなものだが，やはりアーケードゲームメーカーは凄いと感じさせられる。

イメージファイトは奇をてらわない正統派のゲームだ。ビジュアルや色モノに頼ったり，X68000のハードにおんぶした機能誇示もない。海外モノのような最新のゲームコンセプトも持っていない。それでいてやはり面白い。ここらあたりに厳然とした「力」の差を感じる。

イメージファイトはかなりよい移植がされている。X68000への移植ものでは，ある程度「本物そっくり」は当たり前だ。しかし，たまに，どこがどうというわけではなく，プレイしてみて，どこか魂が入っていないと感じられる移植作品もある。そういったものはちょっと見に似ていても人気が出ない。こと，X68000に関しては移植作品にこういった形而上的な要求がされている。完璧に同じではないにせよ，これに応えているというのはとんでもないことなのかもしれない。

総評

とりあえずほとんど不満はない。内容はいわゆるパターンゲームだが，パターンゲームにはパターンゲームの美学がある。問題はコンティニューの方法だ。というのも，最近のX68000のゲームはみんな敵が硬い。わざわざ難易度を無茶苦茶に上げているようなものも少なからずある。しかし，最近の若手X68000ユーザーはそういった超ムズのゲームをあっさりクリアしてしまう。で，ソフトハウスはもっと難しいゲームを作り出す。これがまた難しいゲームは敵が硬いゲームだと思い込んでいるふしがある。困ったもんだ。アンケートハガキを見ると16，17歳近辺に新人類がいるらしい。諸悪の根源である。

で，だ。一般人としては，今後X68000のアクションゲームでは難易度設定の際に生年月日を入力させて「15歳から20歳まではコンティニューなし」とか柔軟に調整してほしい，と，最近結構本気で思っている。

10段階評価

グラフィック	★★★★★★★★★
音楽	★★★★★★
アクション性	★★★★★★★★★
攻略しがい	★★★★★★★★

ひと味違ったシューティングを楽しみたい方に

Yamada Junji
山田 純二

またまたアーケードからの移植作品が登場。このゲームは自機の攻撃力がアップするのではなく，パワーボールというアイテムを使って攻撃していくというもの。もちろん2人プレイも可能，MIDIにも対応している。

ついに発売となったこのゲーム，1987年にテクモから登場したアーケードゲームというのはもう周知の事実だろうけど，X68000版はシステムサコムが移植，開発し，めでたくここに発売となったのだ。

ゲーム自体は縦スクロールのシューティングゲームで，ちと難易度は高め。だからしてもちろん2人プレイもできるようになっている（友達となかよくやろうね）。

アーケード版でもそうだったんだけど，デモ画面を見てまずおやっ？　と思うのは自機のおしりの部分。金魚のフンのようになにやら丸いものがいくつか連なっている。い，いったいこれはっ!?　と大袈裟に驚く人はいないだろうけど，この金魚のフンこそこのゲーム最大（？）の特色なのだ。

金魚のフン，かくしてその正体は！　じゃ～ん，それはこのゲームにおけるパワーアップアイテム，ガンボールの連なっている図なのでした。ここまで引っ張れば大方予想はついていただろうけどね……。

ガンボールこそ我が命

さて，このガンボールとやら，パワーアップアイテムということでご多分にもれず当然いろいろな種類がある（といっても武器関係は5種類なんだけど）。それぞれどういう場面で使用するといちばん有効かがだいたい決まっている。まあ，どれがどんなだっていうのはマニュアルにも書いてあることだし，まず自分でいろいろ使ってみることがいちばんなんだけど。

で，このガンボールを取る手段というのも一風変わっている。特定の敵を倒すと現れることもあるが，主にフリンガーという緑色のサソリに羽が生えたような敵を倒すか，敵から奪い取ることにより得られるのだ。フリンガーの持っているガンボールは，フリンガーを撃つたびに種類が変わっていくので，欲しいガンボールに変えてから奪い取ることもできる。だから自分が欲しいと思うガンボールが出るまでガンガン撃ちまくるに限る。取ったガンボールは，自機の後ろにくっついていき，最高15個までつながる。すごいぞー，15個後ろにでろでろしてて，なおかつ敵さんがわちゃわちゃいたりすると。まるで乱視の検査みたい。

しかし，逆にフリンガーにガンボールを奪い取られることもあるにはある。奪い取ったガンボールをまた奪い取られたり，ガンボールを持っていないフリンガー野郎が，ガンボールのみを奪い取ってそそくさと逃げていってしまったりするのだ。うーん，おじさん一本取られたね，などと不覚にも感心してしまった。

さてさて，やっと使用上の注意までこぎつけたぞ。ガンボールは取った順番につながって，そのとおりに使用されていく。「このシーンではあれを使いたかったのにぃ」などといってもあとの祭り。システムなんだからしゃーないやね。要は，先を見越してガンボールを取る順番，プラス使用のタイミングを考えていけばいいのだ。とはいえキヨタ君じゃないんだから先のことなんてわかんないけどさ。結局何度もトライしたもん勝ちなのだ。

ほかにはなにがある？

さっきも書いたガンボール奪い合いシステムだが，応用編として2人プレイのときにも互いのガンボールの奪い合いができるのが面白い。2人でやれば攻撃力倍増で，戦いは比較的楽になるけど，2人して動き回っているといつの間にかガンボールが相手に取られていたりして，せっかく攻撃しようにもガンボールがない，なんてことも起こる。しかし，この性質をうまく使えば，逆に戦いを有利に進めることもできるのだ。

ガンボールが尽きてしまったときには，相手から分けてもらうこともできる。が，いちばんの使い道はガンボールの組み替えだ。ガンボールは取った順番につながるから，使いたいガンボールが後ろにある場合は，それまでのガンボールを無駄に使うか，それまで我慢しなくてはならない。そんなときには相手にガンボールをすべて渡して，そのあと自分の使いたいガンボールを相手からもらえばいいのである。

グラフィックについては蟲，蟲，蟲の嵐だが，それほどグロいと感じなかった。割と明るめな色合いで，ドロドロ感はない。

X68000用　5″2HD版2枚組　8,800円(税別)
システムサコム　☎03(635)7609

ずらずらと金魚のフンのようにガンボールをつなげちゃえ！

逆にライトすぎると感じたほどだ。ゲーム中で，僕がいちばんグロいと感じたのは，緑色の原色卵を撃つと，でろりん，と中身が飛び出てくるヤツ。あれはグロいよね。

音楽は，なんだか妙。シューティングゲームの音楽は，だいたいにおいてカッコよく，ノリノリのものが多いが，このゲームのBGMは静かめの曲が使われている。特に1面の曲なんかはのどかすぎるほどで，ちょっと気が抜けてしまった。でも，徐々に面が進むにつれて重々しい感じになっていき，はじめのうち感じていた違和感はなくなっていったけど。このゲームはMIDI（MT-32系の音源）にも対応しているので，持っている人はこの妙に雰囲気のある音楽を存分に楽しんでもらいたい。

さて，と，それでは，それぞれの面を見ていくことにしようか。

ステージ解説

1面：のどかな音楽にのって，このゲームのコツをつかむにはちょうどいい面である。敵の攻撃もさほど激しくないので，取れるガンボールは残さず取り，あとあとの面に備えたい。この面のボスは，でっかい襟巻きアザラシ君。攻撃も単調だし，さっさとかたづけてしまおう。あまりもたついていると，華麗な飛び込み技でやられてしまうから，注意すべし。

2面：森の上をがしがし進んでいく。1面で貯めておいたガンボールをうまく使って敵を倒していこう。面の半ばにくるといきなり高速スクロールが始まるが，地形が速くなるだけなので，あわてず冷静に。この面のボスは，穴から飛び出たひとつ目親父2人。目玉を狙って弾を打ち込むべしべしべし。一定時間おきに，ぶっといレーザーを撃ってくるが狙いは甘い。ひょいひょいと死角で待機していればらくちん。

3面：今度は洞窟の中を進んでいくことになる。狭い洞窟内であるから，アイテムによってはそれほどの効果を得られないものが出てくる。障害物で，攻撃力が半減さ

アザラシ君のダイビング，よっけろ～！

このドロドロした暗さがたまらない

れてしまうのだ。3面にはボスキャラはいないが，最後のほうで左右から現れる蔦には注意が必要。撃っても撃っても際限なしに出てくるので，ここは真ん中あたりのいちばん上（変な表現！）に逃げてしまおう。

4面：洞窟を抜けるとそこは嵐であった。派手な雷鳴とともに画面がフラッシュし，なかなかニクい演出である。個人的にはいちばん好きな面だ。敵はうじゃうじゃ，弾もうじゃうじゃ吐いてくるため，もう大変。障害物がないだけまだましだが，このあたりからシューティングのうまい人とそうでない人の差が表れてくるだろう。ボスはぶよぶよした塊。ときどき現れてくる発射口を狙って撃て！

5面：嵐を抜けるとそこは空であった（もういいって）。ここでいちばんヤなヤツは雲のなかから突然現れるコウモリおじさん。とんでもねー，よけられねーぞ，と文句いってもしょうがない。ただただ，よけろーっ。と，ひたすら敵の猛攻をくぐり抜けると浮遊戦艦の登場だ。これの弱点は，左右に見え隠れする砲台である。ここでは，左右どちらかを集中的に攻撃するのではなく，交互に攻撃したほうがいいだろう。

6面：戦艦をやっつけるとそこはジャングルであった（しつこいなあ）。やっとの思いで5面を抜けたと思ったら，さらにキツい面が僕を待っていた。ハンパじゃない攻撃に人食いチューリップ。もう，勘弁してくれの世界である。根性のドット単位のすり抜け技を全開させてクリアを目指そう。ボスは百本足のムカデちゃんだ。

……以下7面8面と続く……だろう。がんばってください（それしかいえない）。

緊張感と面白さの度合い

このジェミニウイングというゲーム，アーケードからの移植とあって，バランスも取れているし，シューテングゲームマニアにとって，ゴキゲンなソフトであるといえる。高い難易度，独特のグラフィック，加えてパワーアップのシステム，どれを取っ

これがウワサの人喰いチューリップだ

ても八分どおり満足できるものだ。

多くのパワーアップシューティングゲームでは，アイテムを取るごとに自機がパワーアップされていく方法をとっている。このようなタイプのゲームでは，途中で死んでしまうとそこからの復活に超人的な技術が必要とされ，プレイヤーは，死んでしまってはもともこもない，という緊張感にゲームを楽しむことになる。

しかし，このゲームはアイテムの攻撃力によりパワーアップしていくのである。さらにアイテムを使うタイミングと取る順番，この2つがよりプレイヤーに刺激を与えているのである。もちろん，前者のような緊張感もあるが，その緊張感にアイテムの使用タイミングという別の要素により，プレイヤーにゲームを面白いと感じさせているといえるだろう。

さて，マニアではない普通の人にはどうであるかというと……，ちょっと難しいかもしれない。最初のうちはともかく，後半戦が異様にキツくなっているから，途中で投げ出してしまう危険性が大。まあ，それなりに楽しめるかもしれないが。

マニュアルにも遅れた言い訳がたくさん載っているけど，そうして時間をかけたぶん十分な出来栄えとなっているので，プレイする側としてはとりあえず安心していい。といったところでサコムさん，次のアトミック・ロボキッドに期待してますよ。

ヘイヘイ！　ジェミニ！（総評）

サコムさーん，コンティニュー3回なんてきつすぎます。せめて，8回ぐらい許してくれなかったのかー（ぐしぐし）。文句はそれぐらいで，アーケードゲーム版との違いは，僕の見た限りないといっていいでしょう。よい出来です。この調子で次回作もがんばってやってもらおう。いけいけ，GO，GO！　サコムさん！

（5段階評価）

難易度	★★★★
グラフィック	★★★
ガンボール	★★★★
移植度	★★★★★

機能満載! 横スクロール

Kageyama Hiroaki
影山 裕昭

デビュー作というものは，作るほうにも買うほうにも「期待と不安」の入り混じった気持ちを抱かせるものですよね。たいていの場合は「不安」のほうが多くなりがちだけど，このゲームは期待してもいいと思いますよ。

新潟のソフトハウス「エグザクト」，といっても知らない人が多いでしょう。それもそのはず，このソフトがデビュー作なんですから。それでも，すでに開発途中の画面写真があちこちの雑誌で紹介されているので，ゲームタイトルくらいはチェックしていた人もいると思います。ちなみにNAIOUSはナイアスと読みます。

どんなゲームなのか

このソフトは基本的には横スクロールのシューティングゲームなのですが，縦スクロールがおりまざっているステージもあります。このときは機首が画面の上を向くのではなくて，機首は右を向いたまま垂直上昇（あるいは降下）していきます。

ステージは全7面で，各ステージの最後にはボスがしっかりと待ち構えています。このテのゲームにはもうなくてはならないコンフィギュレーションももちろんあります。EASY，NORMAL，HARDと3段階の難易度調整ができ，難易度を上げるとアイテムが取りにくい場所に移ったり，敵の出現パターンがいやらしくなってきます。また，ソフト側で連射するように設定できるので，少なくても片方の手は疲れません。ステージ3まで面セレクトが可能，自機は最高5まで増やせ，コンティニューも7回までできるという親切さもいいでしょう。

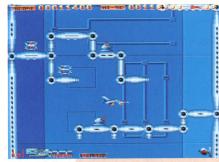
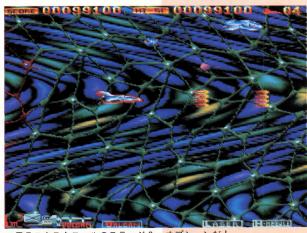
兵器は最初はバルカンのみですが，アイテムを取ることによって，ワイド，レーザー，ホーミングなどが，Aボタンで選択することで使うことができるようになります。それぞれの兵器は2段階までパワーアップします。アイテムはそのほかに，1UPアイテム，シールド，選択している兵器のレベルを最高にするものなどもあります。

また，これとは別に各ステージで味方と思われる未確認飛行物体がオプションを投下してくれます。これにはそれぞれ特長のある4種類があり，攻撃範囲をぐんと広げてくれます。しかし，自機が破壊されると選択していた兵器とともに，オプションもなくなってしまいます。

ここまで読んで「ああ，ありがちなゲームだなあ」と思った方も多いでしょう。しかし，見た目だけで判断するのはよくありません。このゲームの長所はシューティングゲームになくてはならない「スリリングなゲーム展開」が徹底されているということです。画面全体に細心の注意を払ってプレイしている自分にふと気づくほどです。実はこれが簡単そうでいちばん難しいこと。1ステージを通して敵キャラの出てくるタイミングが実によく考えられていて，これが効を奏しているようです。太いレーザーで攻撃してくる敵，ワープしてくる敵，突然後ろから飛んでくるミサイル。どれもこれもプレイヤーの度肝を抜いてくれます。

これこそがシューティングの基本であり，ゲームデザインのセンスが問われるところでしょう。この基本が崩れているシューティングはほかの部分がいくらすばらしくても，意外にあっさり飽きてしまうものです。その点ナイアスのゲームデザインは誉めてもいいでしょう。

ところで，パズルゲームとシューティングゲームが苦手だという（で）君にEASYで遊んでもらったら，ファーストプレイでステージ1をクリアしました。面セレクトできるステージ3までは難易度も低くしてあるように感じます。ただし，ステージが進んでいくと「もうお手上げ」というくらいの難易度にもなります。

業務用のシューティングゲームをかなり意識したソフトなので，ゲーセンに足を運ぶ人はどんなゲームを参考にしているかすぐにわかることと思います。そのためか，ゲーセンにあまり行かない人のほうが夢中になる度合いが強いようです。

ここは縦スクロール。アイテムを取れ！

二重ラスタスクロールのステージ3。オプションだ！

X68000用　5″2HD版2枚組　8,700円(税別)
エグザクト　☎025(247)9160

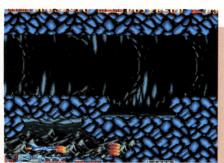
なにが起こったのか

ナイアスは自機の名称ではありません(自機はASPというそうだ)。ナイアスは地球環境の管理，さらには太陽系全体の環境管理を行う推論型コンピュータの名称です。ナイアスの始動直後，地球環境は改善に向かいましたが，その数年後には，人類の活動がナイアスの浄化作用を上回ってしまったのです。そして西暦2143年，あまりにも愚かな人類に対してナイアスはひとつの答えを導き出したのです……。

このゲームの目的はズバリ，ナイアスを破壊することにあります。たとえゲームのなかであっても，地球環境の話が絡んでたりして考えさせられるものがあります。

第1任務は「木星のコロニーを奪還すること」というわけで，宇宙空間からゲームが始まります。最初はほとんど雑魚キャラ。しばらくすると上から巨大な戦艦が……，「えっ，もうボスキャラ？」と思ったら大間違い，これがコロニーです。敵キャラ相手に戦闘を繰り広げている間も，コロニーが上下にフワフワと動いていて，なかなかニクい演出をしてくれます。

スクロールが止まるとコロニーのゲートが開いて，今度は縦スクロールに変わります。ジェット噴射も下方向に変わり，自機が上昇していることが容易に理解できます。そして，コロニー本体に入るとまたまた横スクロール。背景もコロニーの感じがよく出ていて，ガンダムのサイド6を思い出してしまいました。ステージ後半で自機の周りにワープしてくる敵には注意。それを抜けるとボスがやってきます。まあ，楽勝でしょう。

ステージ2は「ASTER」のエンジンを破壊して，冥王星への激突を未然に防ぐことが目的です。途中に出てくるハンバーグみたいな中ボスは安全地帯があるから，それさえわかれば恐れることもなにもありません。最後のボスもそんなに難しくはないでしょう。ステージクリアすると背景がラスタスクロール。思わず唸る演出だけど，この時点で僕はまだこれがなにを表現しているかはわかっていませんでした。

ステージ5は海の中。背景がゆ～らゆ～ら

ステージごとにミッションが解説される

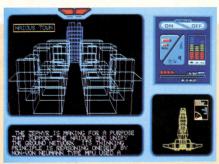
目をひくラスタスクロール

ステージ2は我々の目先を変えるための罠だったのです。「ASTER」の爆発によって作り出されたねじれた空間にASPは吸い込まれてしまいます。あのラスタスクロールは空間のねじれを表現していたのです。そして，この空間を脱出することがステージ3の目的です。

このステージでは2枚のBGが別々にラスタスクロールしていて，とにかくすごい。ハデハデな背景と，ラスタスクロールのおかげで敵の見にくいことといったら……。おまけにそれを保護色として巧みに利用している敵もいます。このステージでは，後ろから突然敵が出現するのでかなりビビります。そしてボスが結構手強い。ま，これ以上ネタをばらすとつまらなくなるからやめときましょう。

だんだんムズく

ステージ4あたりから徐々に敵の攻撃が激しくなってきます。ここは洞窟であちこちで落盤が発生するので，あんまり前のほうには出ないほうがいいと思いますよ。

書きたいことは山ほどあるんだけど，あんまりばらすとこれから遊ぶ人がつまらない思いをするだろうし，難しいところだな。とりあえず中ボスがいます。これがまたすごい。いきなりラスタスクロールして登場するんだもん。そしてレーザーとホーミングを好き放題撃ちまくると，半透明になってラスタスクロール。ワープしてるっ!!

ワープ中はヒットチェックがない。そしてラスタスクロールが止まると，半透明も終わってぐるぐる回りながら攻撃。初めてこれを見て平常心でいられる人はいるのかな。いやいやすばらしい演出です。

ステージ5は海底シーン。今度は海底の建造物が崩れ落ちてくるので，やはり前に出るのは危険です。ここでも海底の様子をラスタスクロールで表現していて，これがなかなかいい味を出しています。しかし攻撃が前にもまして激しくなっている。ジョイパッドしかない僕は右に動こうとして右下に動いてしまったりして，操作もままならない。こりゃ，ジョイスティック買ったほうがいいかな。

これぐらいにしておこう

シューティングゲームの原稿は必要以上に内容をばらしてしまうと，ゲームをつまらなくしてしまうのですごく書きにくい。ステージ6のボスの動きがすばらしいので，ぜひとも見てもらいたいのですが，いったい，そこまでたどり着ける人がどのくらいいることやら。とにかくエグザクトさんのデビュー作は立派。これからが期待できるソフトハウスでしょう。

アイテムに気をとられるとこんなハメに……

ゲームバランスが命

ナイアスは全体的によくできているゲームだと思います。ただ，既存のゲームを参考にしている点が多いので，ケチをつければ，「オリジナリティがまるでない」ということはいえます。それを抜きにしても，ジェミニウイング，イメージファイトといった大作シューティングがほぼ同時期に発売されるので，ちょっとかわいそうな気もします……。

背景のグラフィックはとてもきれいで，これはかなりよろしい。もうちょっと自機をカッコよくしてくれたらもっとよかったんですけどね。なにはともあれ，ナイアスはプレイしていて楽しいゲームなので売れても不思議ではありません。だからこそ，売れたときにはそれで満足せずに「ソフトハウスとしての真価が問われるのは，2本目の出来いかんだ」ということを忘れないようにがんばっていただきたい。なかなかいいセンいってると思いますから。

7段階評価

項目	評価
キャラクタデザイン	★★★★☆☆☆
ゲームバランス	★★★★★★☆
サウンド	★★★★☆☆☆
ラスタスクロール	★★★★★★☆
お買い得感	★★★★★☆☆

敵はどこだ！ ローラーダッシュ！

Saitou Susumu
斎藤 晋

“巨大で重い”戦車はすでに必要でなくなった。そして，NAP(New Age Poweredsuits)が誕生した。ハワード・ボウイ，NAPを操り，戦略上重要なポイント，ブラッディ・アクシスを占拠，破壊せよ。それが指令だ……。

ローラーダッシュをかけると荒れた街並がみるみる画面の外へ流れていく。視界に目標の姿はない。ターゲットキャラクターは残り5機。出てこないなら誘い出すまでだ。が，物陰から雑魚がオレンジの弾を撃ってくる。邪魔なんだよ！　スティックを入れ替えTWINを連射する。

NAPと呼ばれる機動兵器がお馴染みウルフサウンド（MT−32にも対応）に乗ってスピード感，重量感溢れる戦闘アクションを展開する。それがFZ戦記アクシスだ。

これがNAPの威力だ

アクシスの見どころのひとつ，それは画面構成にある。すなわちクォータービューというやつだ。視点を斜めにするとなにがいいっていうと立体感が出る。昔のZAXONなんかも当時は3Dものとして扱われていたように思う。透視図じゃないから遠近感こそないが，クォータービューというのは3次元空間を表現するためのモデル化のひとつだったわけだ。

ゲーム全体は8つの面に分かれ，それぞれが2面構成の合計16ステージだ。それぞれ一定のパターンで現れる将校キャラクターを規定数倒せばステージクリアとなる。

NAPはふつうに操作するときはトコトコと歩くのだが，スティックを同じ方向に2度倒すと，高速移動モードに入る。これがローラーダッシュだ。画面を見てもらえばわかるとおりNAPのキャラクターは画面中央に表示されている。これが高速で移動するといったらどんな感じか想像してほしい。実際に動くのはキャラクターじゃなくて画面のほうである。それもびゅんびゅんの8方向スクロールだ。ローラーダッシュの状態でスティックをコントロールすればちょっとした「めまい」を感じることができる。これがなんといってもこのゲームの魅力だな。また，敵に接近するとNAPは自動的にパンチを繰り出す。すごいぞ！これが強力なのだ。

最初のステージは市街戦だ。目前にタンクが3台迫っている。が，こいつの弾には当たる心配がない。やつの砲台は回らないのだ。ここは背後からローラーダッシュで接近し必殺のパンチを繰り出そう。続いてまた3台。楽勝だ。次は戦闘ヘリがくる。建物の上を飛んでいる間は攻撃ができないから，道路の上空に降りてくるときを狙い撃ちする。でもヘリに気を取られていると思わぬ雑魚キャラの攻撃を食らうはめに。操作法に慣れるまではちょっと苦戦かな。

使える武器は右手のハンドウェポン（Bボタン）と背中のオプションウェポン（Aボタン）があって，最初は各々にNORMALと呼ばれるユニットがセットされている。実はユニットは汎用性があって取り付け場所によって働きが違う。ハンドウェポンとして用いれば弾の威力はさほどでもないが弾数に制限がない。逆にオプションウェポンに装着すると効果は大きいが使える回数に制限がある。

たとえば，BAZOOKAというユニットはハンドウェポンとして使うと，前方のみの連射だが，オプションウェポンとして使うと扇型9方向弾になるといった具合だ。また，武器として使わないときは各部にシールドとして装着される。つまりユニットの残りが耐久度となるってわけだ。

ユニットは標準装備のほかに戦闘のなかで手にはいるものも多い。特にオプションウェポンで使うと楽しいのがいっぱいだ。炎が円状に広がるナパームや誘導弾，残酷なワイヤー攻撃，なぜか爆撃機の援護がくるものまである。しかも，マニュアルにはまだ手に入れたことのないものが載っているではないか。弾数に制限はあるが，せこい考えは無用。ユニットは被弾するたびに失ってしまうのだ。弾切れとなったユニットもシールドとしては有効なのだからどうせなら積極的に使ってしまおう。

さて，多くの敵は弧を描くように攻めてくる。動きが読めれば相手の慣性系に入り込んで簡単に倒せるが，ほとんど平行移動のような関係になるとイライラする。違う方向に撃ちたいときは一瞬ボタンを離してからスティックを入れ直すわけだが，肩に力が入っているとうまくいかない。広いエリアでは向きを変えずにローラーダッシュ

1面のタンク。かわいいけどちょっとマヌケ

強制スクロール面。巨大な装甲車が4台来る

X68000用　5″2HD版3枚組　8,800円(税別)
ウルフ・チーム　☎03(5273)4795

をこまめに利用して敵の動きに追従できるようになると楽しい。

ま，そんなこんなで１面の戦闘ステージをクリア。最後のヘリを撃墜したら周囲はたちまち誘爆の嵐となっておしまいだ。

１面後半のステージは，斜め方向の強制スクロール面。巨大な装甲車と対決だ。攻撃してくる砲台野郎をやっつければいい。２面は原野。３面は神殿，４面は洞窟と続く。

目の前で弾を炸裂する４面のボス

エレベータの面。スピード感，浮遊感に満ちた楽しい面だ

やっぱり拳が頼りだね

僕が一番気に入ったのは５面前半。高速で上昇するエレベータのステージだ。周囲を取り囲むようにヘリが飛来し，前後のエレベータが徐々に速度を合わせてくる。もちろん敵キャラが乗ったやつだ。ここまでこれるようになれば，NAPの操作もそれほど不自由はない。リングの誘導弾も２面のボスキャラでかわし方は修得ずみだ。

後半は文字どおり間抜けな大砲が待っている。あまりの間抜けさに哀れを感じてしまった。哀れな大砲よ，安らかに眠れ。

６面は１面や４面と逆で前半が強制スクロール面。敵は１匹。しかし，巨大な弾丸を一気に吐き出すように撃ってくる。冗談ではない！　と思ったが，これが不思議とよけられるのだ。まるで自分がうまくなったかのような気分に酔いながら，撃破。

後半には巨大な建造物のまわりを囲む２重の動く歩道が。こいつの上ではベルトの向きに逆らって歩いてもほとんど進まない。じたばたしていると思わぬ被弾にあう。

７面。撃破しなければならない将校キャラは25と最も多い。この面では結構新しいユニットが手に入る。とはいえ，ゲームはまもなく佳境であるから，このあとの敵に備えてユニット（シールド）を無駄に失わないようにしたい。後半はどうやら要塞最後の中枢のようだ。まわりをガードする敵の動きがかなり素早くなっている。

さあ，敵要塞内部をことごとく破壊したボゥイの前に最後の敵が立ち塞がる。敵はたった１機の赤い機動兵器だ。しかし，速い！　NAPと同じ運動能力を持っているようだ。回り込もうとしてもなかなか後ろを取らせてはくれない，どころか，うかつに近づくと逆に後方に回り込まれてしまう。反射的にローラーダッシュで後退だ。どうする？　RAYをオプションウェポンに装着。８方向レーザーで死角からの侵入に備える。これなら敵はよけられない。が，そうした思惑もどうやら期待倒れに終わった。敵はちょっとやそっと被弾してもびくともしない。こちらの火力じゃ倒せないんじゃ？　そう思ったとき，しばらく使っていなかった戦法を思い出した。仕掛ける前には一発も食らいたくない。コーナーの狭い通路に赤い影を見つけるやいなや，私はスティックを斜めに入れ，ローラーダッシュで突進していった。

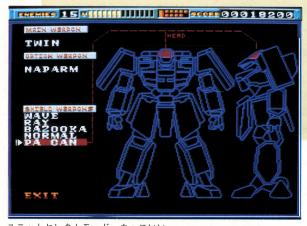
ユニットセレクトモード。カッコいい

お陰さまでエンディング

アクシスは，ステージのバリエーションが楽しめる。強制斜めスクロールの面やエレベータの面はやっていて本当に楽しい。が，市街地や広いフィールド面ではパソコン用ゲームにありがちな冗長さが目立つ。人によっては最初の市街地でほとんどターゲットキャラが出てこない状況に直面する。僕より先にプレイしていた何人かは敵の居場所がわからないと途方に暮れながら街中をさまよっていた。雑魚キャラを30匹ぐらい倒せば出てくるんじゃ？　なんて憶測もあったほどだ。

実は，市街地をなぞるように一定方向に探しても敵は出てこない。若干コツがあるのだが，ちょっと挑発的な動きが必要なのだ。逆に敵の誘い出し方さえわかればゲームは俄然テキパキと進行する。

あと，残念なのは，オプションウェポンが存分に生かせていないことだ。僕の場合，３面で拾えるPA CANをハンドウェポンで使えば（これがかなり強力）あとは特にこれが必要といったものを感じない。ユニークな武器（使って楽しいもの）が結構あるだけにちょっともったいない。

というわけで，EASYモードならアクションのかなり苦手な人でもクリアできるはずだ。ラストステージは「脱出」。ひたすらよけるだけだが，そのままエンディングへとつなげるあたりの演出もいい。よーし，今度はHARDで挑戦しようかな。

この路線でガンバッテ！

X68000にはアクションゲームがたくさんあるが，評価の高いのはアーケードからの移植ものが多い。逆に，X68000で開発されたオリジナルゲームでアーケードに移植しても通用するものといえばちょっとお寒い状況だ。前作グラナダは数少ないそういうゲームだったように思う。このアクシスも基本的なゲームデザインの面ではアーケード指向のゲームだろう。クォータービューとか８方向スクロールというのは必ずしも万人受けしないかもしれないが，ウルフ・チームさんにはこういうとんがった部分でこれからも意欲的な作品に挑戦してもらいたい。

もちろん，グラナダ，アクシスとくれば，こっちだってそれ相応の期待をしなくちゃね。次回作もよろしく。10段階評価

項目	評価
グラフィック	★★★★★★★★
サウンド	★★★★★★★
スピード感	★★★★★★★★★
イライラ感	★★★★★
オープニングデモ	★★★★★★

真の戦場の姿とは?

Yamada Junji
山田 純二

いつもちょっと変わったゲームを発表してくれるアートディンク。今回のこのゲームは戦場をよりリアルに表現するために，いろいろな新しい手法が取り入れられている。さて，その結果やいかに？

「機甲師団」は戦闘画面からHEX画面を排除し，さらにリアルタイムで戦闘が進行していくシミュレーションゲームである。舞台は第2次世界大戦におけるヨーロッパ戦線の地上戦。プレイヤーの最終目標はノルマンディから上陸した連合軍第7機甲師団7大隊を率いて，全12の戦場で待ち受けるドイツ機甲師団を撃ち破り，ヨーロッパ戦線を制圧することとなっている。

シミュレーションというのは，ある事象をモデル化し，そのモデルを操作することによって現実の事象を理解し予測を深める技法である。この場合の現実事象というのは地上戦であり，ゲームの性質上，モデル化するにあたってはできるだけ，より現実に近いものにしなくてはいけない。もちろん，すべての要素を盛り込むのは不可能なので（仮にできたとしても，それがゲームとして成り立つかは疑問），たとえば光栄に代表される一連のシミュレーションゲームでは，登場するキャラクタを現実に近づけるという方法を取るなどしている。

それに対してこのゲームの場合には，HEXによって画面を細切れにすることを廃止した戦闘方式とリアルタイムなゲーム進行を採用することで，プレイヤーにより実戦に近い雰囲気を感じさせようとしている。

はずかしながら僕はシミュレーションゲームというものを，これまで一度もプレイしたことがない。どちらかといえば，短時間で終わるシューティングゲーム専門であった。1プレイに数時間，もしくは十数時間もかかるようなゲームは敬遠していたのである。今回，なぜこのようなゲームを選んで（自分で買った）しまったのかというと**リアルタイム**シミュレーションという，売り文句を聞いたとき，このゲームなら純粋に戦闘を楽しむことができる（あまり時間もかからずにね）のではないか，と思ったからである。

そして，最終目的を達成した現在，その思い込みは果たして正しかったかどうか……は最後に述べるとして，まずはこのゲームの内容を見ていくことにしよう。

X68000用 5″2HD版3枚組 9,500円(税別)
アートディンク ☎0474(77)7541

基本構成

まず，取り込み画像をふんだんに使ったオープニングが始まる。そして，それが終わると作戦内容が表示され，画面は師団司令部（GHQ）に移る。これは部隊全体の作戦行動を指揮するための画面である。

一番最初に行うことは，ユーザーディスクの作成。マニュアルどおりに作成すれば，いよいよゲーム開始である。そうそうマニュアルは60ページと比較的短いので，ゲーム開始前にしっかりと読んでおくことを勧める。

最初は，爆発マークのついているマップAに部隊が上陸しているので，とりあえず戦場に移動。すると，うじゃうじゃと細かいキャラクタがわんさかいる。これがプレイヤーの操る連合軍第7機甲師団7大隊であり，各キャラクタがどのユニットであるかは，図1にまとめて書いてあるので参考にしてもらいたい（なぜかマニュアルには載っていないのでご参考に）。

マップ全体を把握したい場合には「地形図」というメニューバーをクリックするとマップ全体の縮小マップと四角い枠が表示される。この四角い枠が現在表示している範囲で，移動させたい場合にはこの枠をずりずり引きずっていくか，あるいはメイン画面上に表示された格子線をドラッグして（Z'sSTAFFのファイルメニューのように）ずりずりと引きずっていけばよい。また，これらを実行している間は時間が止まっているので，心置きなくマップを眺めることができる。

さて，今度はユニットの移動。これは，移動させたいユニット，または小隊を左クリックし，移動させたいポイントでもう一度クリックすればユニットと目的地が線（移動ルート）で結ばれる。第4の目的地まで指定でき，右クリックすることで確定される。進行中に障害物，あるいはほかのユニットがあるとその場で止まってしまうので注意。また，ユニット自体を右クリックすると，部隊情報を見たりそのユニットに作戦行動を与えることができる。その結果，敵にぶつかると自動的に戦闘開始となる。マップにいる敵部隊を全滅させると（少し間の抜けた音楽が流れて），そのマップを制圧したこととなる。

補給こそ命

戦闘に勝利したからといってほっと一息つくひまはない。今度は補給である。弾薬や燃料の残存量も気にしなくてはならないが，問題は兵員である。中隊司令部から小

部隊の移動先をベクトルで指定

隊の補給は3時間ごとに自動的に行われるが，大隊司令部から中隊司令部の補給は作戦行動によって行わなくてはならないのだ。仮に兵員，物資要請したとしても届くまでには約2時間もかかってしまう。部隊を動かし，実際に戦闘が行われるまでの待ち時間をうまく使って各部隊のチェックを行い，必要ならば補給の要請をしておかなくてはいけないのである。このことをうっかり忘れていると意味のない待ち時間が発生してしまう。リアルタイムにゲームが進行していることを決して忘れてはいけない。

補給に関係してもうひとつ考慮しておかなくてはならない点がある。それは，各部隊にいる士官で，部隊全体の戦力は将兵のいる残存部隊数によって決まるのである。部隊にいる士官が死ぬと部隊が全滅してしまい，戦力が低下してしまうのだ。だから，将兵が死んでしまったらGHQ画面の部隊情報で新しい将兵を任命して，部隊を再構成してやらなくてはならない。残存将兵の数は限られており，あとからは新しく追加できないので，部隊が危なくなってきたら前線から撤退させるか，降伏させたほうがいいだろう。

作戦行動は「砲撃要請」などが中隊司令部か大隊司令部から行うことができるが，小隊は「降伏」しかできないことも覚えておかなくてはならない。

航空部隊を使え

第7機甲師団以外に作戦行動を要請できる部隊として，第3軍所属の航空部隊がある。この部隊は完全に独立しているので，いくら呼び出しても第7機甲師団には影響がないから，非常においしい部隊である。

要請の方法はGHQ画面でメニューバーのスケジュールをクリックし，要請内容，場所，時刻を設定すれば律儀にスケジュールどおりの行動してくれる。しかし，一度にスケジュールを設定すると司令部のほうが勝手に時間調整をしてしまう。一度決めたスケジュールは取り消せないので，役に立たないように思える。しかし，実は抜け道がある。スケジュールをためてしまうと時間調整されてしまうのなら，ためなければよいということ。それだけである。

航空部隊からの爆撃

勝利目前，敵は師団司令部のみ

たとえば，マップBに爆撃要請をしたい，ついでに兵員の補充もしたいという場合には，とりあえず最短時間に爆撃要請を行うようにスケジュールを設定し，戦闘場面に移ってから爆撃されるまで待つ。爆撃が終わったらすかさずGHQ画面に戻り，今度はスケジュールに兵員補給要請を設定する。この場合も時刻は最短時間になるようにする。すると，10分後に設定できる。

操作は面倒だがこれは便利。物資補給は，わざわざ中隊からの補給を待つより，航空部隊からの物資空輸を使えば，一発でマップにいる部隊に補給が完了してしまうので，これまた便利である。

28時間の死闘

以上のことを基本に，ガシガシ戦闘を進めていけば，必ず，勝利を手にすることができる(本当かなあ)。結局，制圧までの総プレイ時間は28時間ぐらいだった（個人差はあるだろうけど)。じっくりと腰を据えればもっと楽しめるだろうが，僕の場合時間的な制約があったので結構キツかった。

最初の思い込みの「割と時間がかからないだろう」というのはちょっと違ったようだ。ここまで時間がかかった最大の原因は，ゲームの進行が遅いということ。ゲームはだいたい現実時間の10倍ぐらいで進行していく（つまり6秒でゲーム中の時間が1分進む）のであるが，もし，ウエイトをかけてわざとそうしているならゲームスピードを自由に設定できるようにすべきである。「それでは忙しすぎてゲームにならないよ」，と勝手に判断されては困るのである（うーんわがまま)。某A.T.氏曰く「プレイヤーをあわてさせるくらいがちょうどいいんですよ」。まったく賛成だ。

文句ついでにもうひとつ。司令部からのメッセージ表示に少し問題があるんじゃないでしょうか。一番下のラインに1行表示されるだけなんて……。ヒストリ機能を付けろとまではいいませんが，せめてウィンドウを開いてよ。しかも，マウスをクリックしてしまうと消えるし，おいらとっても悲しいぜ。

しかし，まあ戦闘のシステムについては気に入りました。たしかに，無味乾燥なHEX画面よりも実際に戦闘をしているという雰囲気はよく出ている。マップのグラフィックも渋く決まっているし，効果音はサンプリングを使っていてこれまたよし。戦闘シーンでの味方の部隊が全滅したときのくやしさ，敵の将兵を倒したときの快感。そして，補給を待っているときのいつ攻撃を受けるかというスリル。これはリアルタイムだからこそ味わえるものであろう。

良くも悪くもアートディンクの個性が光るこのソフト。短気な人にはちょっと辛いかもしれないが，時間的に余裕のある人にはお勧めである。

図1　ユニット表

人間辛抱だ

このゲームにはBGMというものがない。まあ，オープニングミュージックを聞くかぎり，BGMがなくて正解だったかもしれない（ちょっときついかな)。本文でも述べているとおり，プレイ時間はかなりかかる。その間，同じ曲をエンドレスで聞かされてはたまらないかもしれないということである。耳がさびしい人は自分の好きな音楽でもかけて，プレイするのがいいだろう。さすがに，アーシアンのCDはいまいちであった（ああっ趣味がバレる)。それでもまだひまな人は漫画でも読みながら，もしくはプログラミングでもしながらのんびりやるというのがよい（体験談)。あまり長時間ほったらかしにしておくと，どうなっても知らないけど。

5段階評価

戦闘	★★★★★
グラフィック	★★★★★
効果音	★★★
アートディンク	★★★★
総合	★★★★

ひさびさの3Dシューティング登場

Yoshida Kenji
吉田 賢司

あの「メタルサイト」を作ったCROSS-WONDERのスタッフが開発したというオリジナル3Dシューティングゲームが，ビクター音産から緊急発売されました。その技術が今回もうまく生かされているようです。MIDI対応。

異次元空間を中心とした舞台でくり広げられる壮絶な戦いと，MIDI対応（といってもまたMT-32だけどね）の高品質なBGMが最大のウリ。毎日掲示板から「休講」の2字を探すファンキーな大学生も，冬季講習の申し込みに学校を休む中高生も，日夜漢字書き取りに明け暮れる小学生も，ちょいと汗ばんだその手を休めて単純明快のシューティングでもどうです？

ヨカッタね,スコアもついたよ

ゲームは，バイオパワードスーツ「ウィネス」を装着した女性自(マイ)キャラを操作し，前後から迫りくる敵を撃ち落とすというもの。また，4つのオプション兵器のうちひとつを装備でき，これを各面スタート時に選択します。ゲームはエネルギー制を採用，これは面クリア時および一定数以上の敵を破壊すると順次補給され，逆に敵からの攻撃を受けたり前述のオプション兵器の使用によって消耗します。さらに，自機を飛翔させることによっても常時消費されます。これは制限時間と考えてもいいでしょう。

自機は飛行スピードを調整でき，単にスクロールスピードを上げ下げするものにとどまらず，なんと後退もできるのです。この機能により，バラついて出現する雑魚キャラや高速飛来する敵などてこずりそうなヤツは，通常より長い間画面にとどめておき確実に破壊することができます。

そうそう，西川善司氏が「メタルサイト」のレビューのときに「X68000のオリジナルアクションゲームは，なんでスコアがないんだ」と叫んでいましたが，なんでも彼の意見を尊重して今回は，そういった表示関係を特に大きくした，ということです。善さんヨカッタね，スコアついて。

これがエリアⅠ，異次元空間だ！

各ステージをちょこっと紹介

ゲームをスタートすると，一面真っ赤の亜空間。ほとんど練習同様なので，ボス出現までスピードを最高速に保ってバンバン雑魚キャラを落としましょう。このステージでは特殊兵器を装備する必要はないでしょう（そうそう，特殊兵器を装備しなければ，面スタート時に150ポイントのエネルギーボーナスが加算されますよ）。

ステージ2は一変して近代都市へ。このステージもステージ1同様ラクにクリア。

ステージ3から少し難しくなってきます。高速飛来する堅い敵は，スピードをやや後退気味まで落としてから攻撃するとよいですね。ボスキャラが結構堅いので，この辺から適当な特殊兵器を装備するといいかもしれません。ステージは全部で10ありますが，コンティニューはステージ6まで。頑張って最終面を目指しましょう。

最後に

確かにデキは「まずまず」だし，ゲームも楽しいのですけれど，どうもキャラクターが……。敵キャラはなんかデコレーションケーキみたいにノッペリしているし，特に女性自(マイ)キャラはせっかく「人間」なのに「スペースハリアー」の主人公のような人間臭さが感じられないのが少し残念。

お相手は，約1年ぶりの再登場の吉田（みんな覚えてないだろーな）賢司でした。

面をクリアするとこんな感じで次のエリアへ……

4面のボス，ホーミングをよけるのもヤッカイだ

技術を感じるプログラミング

「ニューラルギア」は，なかなか「技術力」を感じさせてくれます。まず，ディスクアクセス。オプション兵器選択時に，どうも並行してディスクを読んでいるようなのです。うーむ，まるでバックグラウンド処理のよう。あと，面クリア時のフェードアウトの仕方の美しさ！ 写真を見てもらえればわかると思うのですが，背景や自機の動きをそのままに，エコーをかけてホワイトアウトしていくんです。うーん，この開発メンバーには，今度はアーケードゲームかなんかを移植してもらいたいな，ディスプレイ縦置きモードかなんかもついた奴を（思い始めるととどまるところを知らない吉田賢司)。期待しているのでがんばってください。

7段階評価

項目	評価
ゲーム性	★★★★★☆☆
操作性	★★★★★☆☆
BGM	★★★★★☆☆
グラフィック	★★★★☆☆☆
技術力	★★★★★★☆

X68000用　5"2HD版2枚組　8,800円(税別)
ビクター音楽産業　☎03(423)7901

早かったじゃん,完結編

Komura Satoshi
古村 聡

サコムのノベルウェア，闇の血族の完結編がほぼ予定どおり発売，続編にしてはなかなかお早い登場でありますね。今回の舞台は予告どおりメキシコシティ。そして，そこでも魅由の周りで奇怪な事件が……。

舞台はメキシコシティへ

1990年初夏，新宿。ひとりのファッションモデルが変死体で見つかった。そしてまたひとり犠牲者が……。殺された2人と同じスタジオに所属していた魅由は，彼女らの死に不審を抱き独自に調査を始めた。そして魅由はその答えを求め親友の里沙とメキシコシティへと飛び立ったのだった……。

里沙は疲れているらしい。どこか抜けているような，変な虚脱感に包まれている。

「怖い……のよ———」

里沙が小さくつぶやいた。

そして里沙が消えた。魅由はシャワーを浴びていた。バスルームの床が一面血の色になった。シャワーから血が!? 直後，里沙の悲鳴。あわてて魅由が部屋に戻ると，もうそこに里沙の姿はなかった……。

魅由はティオティワカンの遺跡にある太陽のピラミッドにきた。影のない南中時のピラミッド。音が消えた。人々のざわめき，喧騒が。人々は背後にいる。音だけのない世界。いや，そこは自分以外の音のない世界，だった。そしてそこに魅由は信じられないものを見た。そこに現れたのは……！

X68000用 5″2HD版4枚組 8，800円(税別)
システムサコム ☎03(635)7609

さて,ゲーム周りは？

こんな感じで話は進んでいくわけですが，2，3感想なぞ。まずシナリオについて。前編とくらべるとずいぶんとコ〇ル〇調というか新〇〇子モドキ色というのか，

「はふ，ちょっとためいき」

「んーっもう，JESUS」

はなく……はないですが少なくなった（私が慣れてきてしまっただけだろうか!? 考えたくないなあ）ような気がします。

さて，システム。この闇の血族もほかのノベルウェアシリーズと同じようなシステムなのですが，ノベルウェアもプログラム的にはずいぶんといろいろ改良されているようで，特に音楽はさすがサコムという出来。いやぁ，MT-32を使うとほとんど映画音楽だもんなあ。うーん，はりぃうっど。

それと画面効果。こいつはっ，前編では拡大縮小なんかやってくれちゃって，へへーってなもんだったのですが，完結編はもっとびびりますよぉ。なーんとあの，

ラスタースクロール

までやってしまっているのだ!! やってくれるじゃん。アーケードのアクションゲームの移植でもないのに（まあ，ノウハウはあるだろうけど。ノベルウェア以外にもいろいろ作ってるしね）こういうことを平気でやってくれちゃうのには……いやいや，まいりましたっ！

この闇の血族も前編だけの状態に比べると，ずいぶんよくなったのではないかと思います。話がまとまっただけでもね（これが本当のオチついた，なんつって……ああっ)。なんだかんだいっても，シナリオもクライマックスとあの最後のエンディングで，結構感動する人もいそうですし。まあノベルウェアとしてはなかなかよくできたほうになるのでしょうか（私は「ソフトでハード……」の2がいちばん好きなんだけど)。

◀ついにメキシコにやってきた魅由

▲お!? これは!?
アニメ効果だったりして

ノベルウェアのたどり着く先は？

まず最初に。これはゲームではなく“ページをめくる必要のないBGM&絵つき小説”です。確かにたまにマウスを使いますが，その行為自体は“ゲームを解くため”ではなく“魅由になるため”のものです。次にシナリオですが，これ自体のセンスはわりといい。いま流行の覚醒ものだし。ただ，魅由の行動と思考で成り立っているだけに，どうも先走りや考えすぎのところが多く，見ているとめくれないページを戻したくなる衝動にかられます。さてグラフィック。背景はともかく，肝心のキャラクターの表情がちょっと乏しい。謎を解くためにメキシコまでやってきた行動力のある女の子にしては，曖昧な表情が多すぎます。まあ，それでもシステムの進化や雰囲気作りのよさも手伝って，全体的には合格点といえるでしょう。（出口香）

	（10段階評価）
グラフィック	6
音楽	9
センス	8
ノベルウェアシステム進化度	9

今年のトレンドはスポーツするヤンキーだ

Ogikubo Kei
荻窪 圭

またまた登場のくにおくんが今度はサッカーに挑戦。もちろんルールはなきに等しい。スナオにアクションとして楽しもう。しかし「不良更生＝スポーツで発散」という図式はゲームにまで浸透するんだな……。

泉大介氏に触発されたわけではないが，今月はEOS-1000を衝動買いしてしまった荻窪圭である。本当はEOS-10を買いたかったのだが，まあ，いいや。PIXYといいEOS-1000といい，はたから見ると「よくそんな金があるなあ」と思われる方も多いだろうが，金はない。あるのはクレジットカードである。お間違いなきよう。

ああ，EOS-1000って，安くて高機能なのはいいけど，本当に玩具だね。大事なところ以外はとにかくそこらじゅうプラスチックで作ってあって，軽すぎてかえってホールドしにくい。日本ってのは，玩具みたいな製品を作るのは得意だね。

ゲームもそう（いきなり無謀な展開だが）。大作やらスケールの大きいゲームは演出文化や大河文化で先を行くアメリカには勝てないけど，玩具みたいなゲームは本当に得意だ。くにおくんシリーズなんか，その典型。ああいった中身のない単純で簡単なゲームを作るのはうまい。シリアスでかっこいい演出をさせると日本のゲーム業界は（映画業界も）ぼろぼろだけど，コミカルで軽い演出はなかなか。アメコミと少年ジャンプの違いかもしれない。

X68000用　5″2HD版2枚組　8,800円(税別)
シャープ　☎03(260)1161

あ，そういえば，先月のスタジオボイスの海外コミック特集を見たのだが，フランスのメビウスって人の絵は凄い。描き込みが凄い，っていうのなら大友克洋も凄いが，このメビウスって人のは色使いやセンスも凄い。そのぶん，1コマひとコマのスピード感はないけど，スピード感とノリだけで描かれているような日本のコミックを読み慣れてると，こういうのが新鮮でいいね。

で，ここで紹介するゲームはジャパニーズ丸出しの，スピード感とノリだけが勝負のゲームだ。

＊　＊　＊

いま知ったのだが，サッカーってのは，アソシエイション・フットボールの略語で，アソシエイションを短縮した語に“er”をつけたものなんだそうな。「研究社 英和中辞典」のSOCCERを引いたらそう書いてあった。初耳。

さて，サッカーといえば，ワールドカップ。マラドーナなんてどうでもいいけど，私は，あのごついキーパーとブロンドの長髪をなびかせて突進するFWが好きだったんだよな。何が悔しいって，あの長髪FWが決勝戦に出られなかったこと。うーん，気に入らん決勝だった。

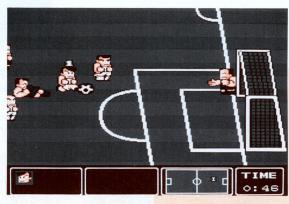
で，このサッカーゲームである。間違っても「熱血高校サッカー部」ではないぞ。「熱血高校ドッジボール部　サッカー編」だ。ドッジボール部の部員がサッカー部マネージャーの“みさこ”に無理やりくどかれてサッカーするという，いわば，他流試合なのだ。ドッジボール部だって，美人マネージャーの色香にはへろへろなのだ。私だって色香には弱い。絶世の美女に色気で迫られたら，一太郎だって使っちまうぞ。

さて，このみさこってのがまた曲者なのだ。なんてったって，「私の（ピーッ）なんてどうでもいいのね」なんていって，女を武器にドッジボール部にサッカーさせようっていうのだから。ちなみに，ピーッって書いたのは，よい子のOh!Xには書けないような言葉だからではなく，本当にピーッっていうからだ。なんか，サンデーやマガジンに出てきそうな女マネージャーだ。

オフサイド，な〜んやそれ？

他流試合とはいえ，ドッジボールより面白く遊べた私である。なんでか。

ドッジボールは，画面上で一度に動くキャラクタがわんさかいるから，ときどき処理が重いのか反応が鈍いときがあったが，サッカー編ではそんなことはない。なんてったって，このサッカー，室内サッカーでもあるまいに，6人制なのだ。キーパーとDF2人と，MF1人と，FWが2人だけなのだ。おうおう，これは楽でいい。そんでもって，動かせるのはMF（ミッドフィルダー）だけ。ポジションは変更できるから，くにおくんじゃなくてもいいわけだ。

そもそもサッカーで重要なのは個人技だけではなく，連係プレーであり（たとえば，翼くんと岬くん，薬丸とシンゴみたいな関係だ），作戦だ。それがだな，パス主体で攻めるかドリブル主体か，タックルするか，自分のキャラ以外にシュートをさせてやるかどうかなんて試合前とハーフタイムに設定できるのだ。そのうえ，自キャラがボールを持っていないときには，“パスしろ”だ

くにおドリブルするの図

の“シュートを決めろ”だのの指示は出せるし，相手ボールのときには“さっさとタックルしやがれ”なんていえるのだ。これは楽しい。

味方の動きのアホさ加減にうんざりしたときは指示を出せ，だ。

ほら，なんか，まともなサッカーゲームのような気がしてきたでしょ。まともなサッカーゲームの素質はもってるよ。しかし，痩せても枯れてもくにおくんなのである。例によって人知も物理法則も越えた必殺シュートはあるわ，オフサイドはないわ，がんがんタックルしてもイエローカードは出ないわ（それ以前にファールがない）という超アバウトなサッカーなのだ。

▲ハーフタイムにうんこすわりの図
◀ダイビングヘッドシュートの図

演技の下手なみさことは

まあ，例によって，脳天気なストーリーがある。サッカー部員が食中毒になったから代わりに出場してくれっていうことだ。みんな，いやがってるんだが，みさこの「ピーッしてあげちゃう」のひと言でみんなの目ん玉が“抜作先生”になってしまう（古いなあ）ところもありがちだが，これがまた本当にPCMでしゃべっちまうところなんかが「闇の血族」より凄いぞ。

で，だな，ときどき食中毒の治ったサッカー部員が復帰してきたりする。これがまた，目立っていい動きをするでもない気安め野郎ばかりなのだ。でも，途中でメンバーを変えられるというのはルーチンワークと化しがちな展開に刺激を与えていいぞ。

それでまたいちいちしゃべるんだ。PCMで。これがまた，下手くそで下手くそで。プロの台詞棒読み屋とみた。こんな味は素人にはなかなか出せない。なかなか味わい深いので，1度はちゃんと聞いてやろう。みさこのタカビーぶりも腹が立っていいぞ。

試合は至極簡単で，至極面白い。ホイッスルで試合が始まる。まずパスし，くにおくんは指示を出しながら，FWの2人にボールを持ち込んでもらうのがいいだろう。

レバーを倒すだけでドリブルしてくれるし，適当にBボタンを押せば適当な味方にパスするし（適当なぶん，よく，インターセプトされるけど），Aボタンを押すと，シュートしてくれるので簡単。傑作がA＋B。走りながらパスされるボールに合わせてA＋Bを押して，ダイビングヘッドバット！　じゃなかった，ダイビングヘッドシュートなのだ。これがまた，必殺「なめんなよ（ってしゃべるんだもん）」が出やすいので重宝だが，タイミングが難しい。空振りして寝転がってるくにおは情けないぞ。

止まった状態でA＋Bだと翼くんばりのオーバーヘッドシュート，これがまた気持ちいい。はずすとただみっともないだけなので，何度もくるくる回りながら，「ジャニーズでーす」と馬鹿のふりをしてごまかそう。

基本的にパスを受けるタイミングが結構難しいのだな。ノートラップのボレーシュートもできるが，これも結構難しい。

問題は，敵チームだな。トーナメントモードで遊ぶ。対戦するのは全部で12校だ。そう，12校。考えてもみろ。12回も戦わなければ優勝できない大会って，いったい何校参加してるんだ？　もの凄い数だぞ。

まあ，それは色物アクションだからいいとして，敵だ。4回戦で当たるマタギ学園なんて，凸凹の土地で，石にけつまずいて転びながらの試合だし，8回戦の恐山商業高校なんて，氷の上だぞ。滑る滑る（がんばれ受験生！）。

ハーフタイムショーもなかなか楽しい。吉本工業高校の落語とか，一本釣り水産高校の釣りとか，なかなか笑える。

この一本釣り高校ってのがまた卑怯な必殺シュートを使ってくれるんだ。なんてったって，サッカーボールがおサカナになっちまうんだ。いくら優秀なキーパーでもかつおは取れないぜ。

攻略法？　あったら教えてもらいたいもんだ。ゴールキックを蹴ろうとするキーパーの前でゴールキックを止めて（間抜けなキーパーなのだ），そいつを蹴り込むとか（たまに成功する），味方にシュートを打たせて，キーパーのはじいたこぼれ球を蹴り込むとかいろいろ手はある。しかし，準決勝の服部学園には通用しないのだ。いくら忍者だからといって，石にけつまずいても転ばないとか，足が異様に速くて追いつけないとか，簡単に必殺シュートを打ちやがるとか，とにかく気に入らないのだ。

2人プレイもあるでよ

例によって2人プレイもある。対戦もできるが，ここはやはり力を合わせてトーナメントを勝ち進んでいきたい。2人プレイだとプレイヤー1がMFを，プレイヤー2がFWのひとりを担当するので，ちゃんと練習すれば息のあった攻撃が可能だ。互いに罵倒しあいながらゲームを進める面白さはちゃんとある。

くにおくんシリーズは，ゲーム専用機の世界では“1本でいろんなスポーツが遊べる”やつもあるらしい。そういうので気楽にお遊びできるやつも出してもらいたいね。

みさこに哀願されるの図

総評

とにかくくにおくんシリーズである。操作の反応はいいし，動きもいいし，コンピュータは（味方も敵も）馬鹿だし，というコミカルアクションの王道を行くソフトである。

飽きたら，ポジションを変えて，くにおをキーパーにしてみたり，変な奴を担当したりして楽しめる。一応，みんな，必殺シュートを持っているので，それを見出す楽しみもある。

こういうゲームばかりになっても困るけど，アクションゲームが得意ではない私でも1本は揃えておきたいソフトだ。ドッジボールよりおすすめ。V'BALLよりおすすめ。ってとこだな。ワールドコートよりおすすめ……とはいえない。

5段階評価だがや

キャラの反応度	5
キャラの頭のよさ度	3
みさこのタカビー度	5
学芸会度	5
暇潰し度	5

純国産本格派模擬飛行ゲーム

Nishikawa Zenji
西川 善司

特異な前進翼構造を持つ，特殊戦術戦闘攻撃機“MI-C.A.D.O.”。この戦闘機を操り，空中戦，軍事施設への対地攻撃，水上艦隊への対艦攻撃，さらには偵察，艦船護衛任務などのミッションを遂行せよ。

アメーリカのコンピュータ野郎は，そうとうなフライトシミュレータ好きらしい。

「パソコンを買ったら作ってみたいゲームは？」
なんて質問をアメーリカのパソコンショップでAMIGAなんかに齧りついてる小僧に浴びせたら，十中八九が「フライトシミュレータさ，ベイビー，ファッキューメーン」とか答えるんだろう。

さて，もしこれを我が国ジャパーンの人間に聞いたら，何と答えるんだろうね。口を揃えたように，「○ースみたいなアクションRPG」とか「○ラデ○ウスみたいなシューティングゲーム」なんていう答えが返ってくるのが目に見えてる……。

好きだというだけあって，舶来フライトシミュレータはよくできている。画面がガシガシウニウニ動くのなんの。視点は変えられるわ，道に車は走ってるわ(F-29)，オートパイロットが賢いわで(Falcon)，パソコンで動いているとは信じたくないほどの凄さ。

さて，今回発売された「遊撃王II」。アメーリカのフライトシミュレータをかなり意識した作りになっている。さぁ，ジャパニーズの底力を見せてくれるのか，遊撃王II!?

Let's fly！（道の真ん中で大声で言ったら実に恥ずかしいセリフだな，これって）

X68000用 5"2HD版2枚組 8,800円(税別)
システムソフト ☎092(752)5278

洗練された操作系

まず，ジャパニーズにフライトシミュレータのウケが悪いのは，おそらくその操作系の複雑さに起因しているのだろう。舶来のフライトシミュレータは機能が多いだけあって，日本語ワープロ以上の操作キーがキーボード上に割り当てられており，はっきりいってよほどこの筋のゲームが好きでないとやる気が失せてくる。

たとえば，車輪の出し入れ。離陸したあとも車輪を出していると速度が出ない，燃料効率が悪い，挙句の果ては車輪が壊れてしまうなど（よく言えば）細かいところまで凝っている。こういった面倒臭い操作をアメーリカンはきっと「きゃあきゃあ」いいながら嬉しがるんだろう。

で，「遊撃王II」は，というとそういった面倒臭い操作は一切排除されているから，この筋が初めての人も安心安心。一度飛び上がったらプレイヤーは空中戦に集中してればいい。よほど変な行為をしない限り「失速」もしないから宙返りや垂直上昇なんかをやってもOK。

そうそう，もちろんサイバースティックにもバッチリ対応しているぞ。レーダーレンジの切り替え以外はキーボードを必要としないから，ほとんどこれひとつで遊べてしまうのだ。ちなみに私は全20ステージをサイバースティックでクリアした。

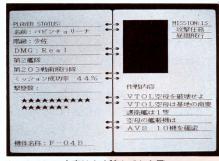
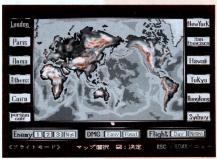
2つのモードで楽しさ倍増

「遊撃王II」は大きく分けると2つのモードで構成されている。「ミッションモード」と「フライトモード」の2つだ。

「ミッションモード」は今回の主役である近未来架空戦闘機“F-04B MI-C.A.D.O.”の開発された西暦20××年を舞台にしたリアルなミッション遂行型のゲームだ。与えられた作戦をうまく遂行できればその功績に応じた階級と新たな任務がプレイヤーに与えられる。

「フライトモード」は“F-04B MI-C.A.D.O.”で，世界中の好きな場所の好きな時間帯のフライトを楽しめるモードだ。こちらは敵機の有無や攻撃を受けたときの操作系統への影響などを自由に設定することができるので「ミッションモード」をする前の飛行訓練にもいいぞ。

また，ロンドンでは「タワーブリッジ」の下をくぐったり，ニューヨークでは「自由の女神（らしきもの）」の上空を飛んだりして，もう気分は世界一周旅行だ。

さて，「ミッションモード」では「ユーザーディスク」なるものを作成しなければならない。RPGのようにゲームの途中経過をこのディスクに記録することができるわけだがひとつ注意しなければならないのがダメージモードの設定だ。これは敵機からの攻撃を受けたときのダメージの操作系統への影響の有無を設定するもの。前述の「フ

ミッションの内容はよく読んでから飛べ

好きな国の好きな町をここで選ぶ

ライトモード」ではプレイするたびに自由に設定できたが「ミッションモード」ではプレイヤー名登録時で一度設定してしまうと途中で変更ができないのだ。

これは純粋に通常のゲームの「EASY」と「HARD」のようなランク設定に相当するので自分の腕前にあわせて慎重に選ぶように。私は「REAL」(つまりは「HARD」)をお勧めするぞ。ダメージを受けるたびに操作系統がイカれていくのがリアルだし，この状態で無事に帰還したときの充実感といったら，スポロガムの絵をうまく切り出せたときの快感に通じるものがあるぞ。

敵艦隊発見，対地ミサイルで攻撃だ！

着艦寸前！　緊張する一瞬だ

多彩な内容の各ミッション

ミッション1はオーソドックスな「敵機を全機破壊せよ」といった内容だ。単純な内容とは裏腹にミッション1は実質的に一番難しい作戦なので，ここでもう断念してしまう人も多いのではないだろうか(破壊すべき敵機の数は8機，自機のミサイル搭載数は4発。つまり，最低でも4機は機銃で撃ち落とさなくてはならないのだ)。でも，あきらめてはいけない。このゲームが面白くなってくるのはミッション2以降なのだ。

その先のミッションを紹介しておこう。ミッション2は敵基地の偵察任務。これは敵領空へ低空より侵入し，敵基地の写真を撮影し帰還するといった作戦だ。ミッションは敵機を撃ち落とすだけではなく，こういったリアルな内容のものがある。しかも夜間飛行。[R]キーまたはサイバースティックのセレクトボタンで画面をワイヤーフレームに切り替え可能なのだが，これが暗視カメラみたいなノリをかもしだしていて，なかなかグーな演出。

そして，次のミッション3はミッション2で偵察した敵基地の破壊作戦。うーん。知らず知らずのうちにプレイヤーを引き込んでいく，言葉で語らずして張られた見事なこの伏線。

さて，この先にも多種多様な内容のミッションが用意されているぞ。たとえば，味方輸送機の護衛任務とか，敵艦隊の殲滅，自機に優るとも劣らない速さで飛来する敵新型戦闘機の撃墜などなど。まあ，あとは自分の目で確かめてくれや，健闘を祈る！

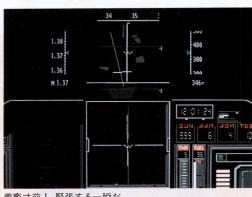
これが護衛すべき機体。あとでこいつが……

着陸，着艦のテクニック

というわけで，無事にミッションを果たせば基地に戻らなければならないのだが，結構着陸が難しい。せっかく作戦内容を果たしても，ちゃんと基地に帰還できなければ元も子もあったもんじゃない。そこで，私，バビンチョ西川の着陸テクニックを伝授しよう。

まず，当たり前だがランディングモードにする。ビーコンを参考に適当な距離を残して機体を滑走路に平行にする。速度は800km/h以下，高度は600m以下にして機首をやや下に向ける。まずは下準備といったところだ。

滑走路が見た目にかなり大きくなってきたら速度400km/h以下，高度は300m以下にする。そして，滑走路へ進入しそうになるころには速度200km/h，高度12～20mになるようにする。

滑走路へ進入したら機首は水平くらいに合わせる。あとはエアブレーキをかけて一気に速度を落とす。そう，高度を操縦桿でコントロールせず，速度を落とすことによって失速させ高度を下げるのだ。この方法は，特に走行距離の短い空母へ帰還する場合に有効だぞ。なぜなら着艦したときにはすでに速度が十分落ちているため短い制動距離で停止できるからだ。

気になる処理速度は？

このテのゲームで一番気になる点が処理速度だ。「遊撃王II」は残念ながらお世辞にも処理速度が速いとはいえない。ワイヤーフレームモードにすると若干速く感じるがそれでも十分とはいいがたい。256×256ドット画面などの低解像度モードを使用するなどしてもう少し高速化を図ってほしかった。

また，動きが多少粗い。DōGAのCGAやアメーリカのフライトシミュレータを見てもわかるが，このテの3Dものは絵の解像度よりも動きの細かさとスピード感が重要なのだ。

気になる点は本当にそのくらいであとはゲームバランス，敵の賢さ，操作性，グラフィック，どれも非の打ちどころのない出来となっている。ゲームの面白さは私が保証するからぜひ買ってじっくりと遊んでみてほしいな。

PC-9801版との相違点

1)　まず，第1にレーダーが格段に見やすくなった。PC-9801版では大地図の上に自機や敵機を表す点がポツポツと動くだけだったがX68000版では表示を拡大縮小できるジャイロコンパスのようなものになり，敵や滑走路までの距離などを把握しやすくなった。

2)　第2に画面左下に敵機映像が表示されるようになった。これは自機の視点から見たもので，これによって敵機との相対的な位置を一瞬で判断することができるようになった。

3)　第3に視点の変更が可能となったこと。カーソルキーで4方向に視点を変えることができる。2)とあわせて使えば敵機を効率よく追うことができるぞ。

4)　アナログスティックでの操作が多少難しくなっている。これはスティックに「遊び」が存在し細かな操作を困難にしているのが原因だ。まあ，これは慣れればたいした問題じゃないかも♡。

(7段階評価)

ゲーム性	★★★★★★★
スピード	★★★☆☆☆☆
サウンド	★★★☆☆☆☆
グラフィック	★★★★☆☆☆
操作性	★★★★★☆☆
お買い得度	★★★★★★☆

きもちいいアクションゲームくださいな

Urakawa Hiroyuki
浦川 博之

トリトーンシリーズの悪玉，バルーサの魂が復活した。主人公スティルはそんなこととは知らず，魔物に襲われたエデルの姫ティアの話を聞き，ティアの兄を捜しに出かけた。なんでもいいから，剣をびゅんびゅん振って，先に進め！

ある日のソフトショップ「うりゃかわ」
いらっしゃいませ。ゲームをお求めですか。ああ，アクションゲームを買いに。そうですねえ，「サイバリオン」なんかもいいけれど，私のお勧めはこの「バルーサの復讐」かな。ザインソフトの新作だ。おや，どうしたんだい，いきなり泣き出して。よし，それならゲームを見せてあげよう。またまたサイドビューのアクションゲームだけど，キャラクターも大きいし，動きも複雑だよ。演出も凝ってる。まあ，そういわずに。テストプレイさせてあげるから座って座って。

展示発表試遊会

まず，デモから見るかい。はい，デモディスクとAディスク。プレイのときはAとBのディスクを入れれば，いちいちデモを見なくていいから便利だよ。それにこのデモは512×512モードを使ってるんだ。枚数は少ないけど絵もきれいだし字も読みやすいだろう。

デモが終わったらスペースキーだ。「ディスクを入れ替えてリセットしてください」だってさ。リセットってのが情けなくていやだ？　細かいところにこだわるねえ。遊ぶ前からそう文句をいうもんじゃないよ。じゃ，リセットするよ。

ほら，（しゅわっ）このメニューの字の出方がカッコいいだろう。拡大縮小するんだ。コンフィギュレーションで（しゅわっ）ディスプレイモードや難易度も選べる。ゲームスタートを選んで。

ほら，1面は船の上だ。サンプリングの嵐の音がいいだろう。揺れる船！　逆巻く波！　BGMはユーロビート調だ。早くゾンビを倒して。しゃがみ突き，ジャンプ斬り，ジャンプしながらの下突き攻撃もできるよ。下突きがいちばん強力だ。おお，うまいうまい。きれいに乗り切った。ここでいきなりボスキャラ大ガニ君が登場だ。沸きだす子ガニ君の攻撃にめげてちゃいけない。大ガニ君の目玉を狙って下突きだ。ああ，なかなか早く倒したね。1面クリアだ。ここでビジュアルシーンがあるよ。

……うふふ。このビジュアルシーンがまたザインらしいんだ。ははは。あははは。君だって，あはは，ウケてるじゃないか。このセンスがさすがザインだろう。

2面は街中。蛙スライムや怪物剣士が相手だ。あ，その灰色の玉は体力ダウンアイテムだから取っちゃだめだよ。ああ，だめだっていってるのに。タイミング的に取ってしまいやすいんだけど，この敵はこのアイテムというパターンを覚えれば大丈夫。特殊アイテムや新しい武器も手に入るからいいじゃないか。さっきのサルが出てきたよ。ここは謎を解かなくてはならないんだ。ビジュアルシーンを見たからわかるだろ。そうそう。

じゃ，もういいかい（プチ）。

毎度あり

どうだい，面白かっただろう。まだまだこの先，奇想天外な展開があるよ。隠れステージもあるし。まあ，ちょっとキャラクターの動きや背景の絵の作りがぎこちないところはあるけど。そうだね，自分の武器の威力ももうちょっと強いほうがいいかもしれない。しかし，マップの構造も考えてあるし，敵をガンガンやっつけるっていうアクションの基本的な面白さでは十分合格だろう。いろんな武器のどれが有利か見ていくだけでも楽しいもんだよ。さあ，どうする？　買うの。そうか，ゲーム中は結構楽しんでたものね。はい，じゃあこれ。どうもありがとう。またどうぞ。

X68000用　5"2HD版3枚組　8,800円（税別）
ザインソフト　☎078(242)2855

こんなカニとも知り合いとは魔王も顔が広い

ゾンビと亡霊の複合アタック

店主うりゃかわ氏のメモ

アクションゲームとしての仕上がりは合格点。下突きでがしがし敵を倒すのが気持ちいいし，ザインならではのデカキャラも魅力。アーケードからの移植などに比べるとまだまだあちこちで見劣りがするが，個性がその分をカバーしている。なにか（いい意味での）B級映画のような不思議な魅力のあるゲームだ。

項目	評価
操作性	7
ゲームバランス	5
音楽	8
グラフィック	5
ザインのセンス	10
熱中度	8

（10段階評価）

スプレッドシートの精神

なにかにつけKamikazeを立ち上げるという荻窪氏。画面に広がる1枚の紙は心掛けしだいで自由なワークスペースとなる。表集計＝会計といった堅苦しい事務用途のイメージにとらわれず，スプレッドシートの醍醐味を味わってみたい。

Ogikubo Kei 荻窪 圭

「大人のためのX68000」も3回目になる。そろそろ本稿が「仕事のためのX68000」でも「企業のためのX68000」でもないことを意識していただけるのではないかと思う。パソコン自体を楽しむのもまたおつだが，パソコンを使ってする行為を楽しむのもまた面白いもの。

そう考えると，日本はまだまだ(X68000に限らず）遅れているように感じる。パソコン自体を楽しむアマチュアプログラマやマニアと呼ばれる人々と，仕事のためにパソコンを使うビジネス屋さんはたくさんいるが，パソコン文化を担うべき「パソコンを使って何かしてみる」人々の盛り上がりがいまいちなのだ。

だから，市販ソフトをうまく使って，その設計者が「あ，こんな使い方もできたのか」と驚くようなことをしている読者がいたら，ご一報いただけるとうれしい。

＊　＊　＊

本場アメリカでは表計算ソフトとはいわない。スプレッドシートという。ここのあたりが非常にアメリカンである。スプレッドというのはSPREADと書き，広がるとか広げるといった意味の動詞である（パンに塗るものという意味もある)。シートは1枚の紙だから，スプレッドシートというのはそのまま訳すと，“広がった紙1枚”ということになる。

つまり，目の前にあるのは“表計算”するためのものではなく，ただの“広がったシート”にすぎないのだ。そのとおり，スプレッドシートは計算にとらわれず，好きなように使ってよいたくさんの升目にすぎないのだ。このあたりを押さえておかないと，“表計算ソフトはビジネスソフトであるから，普通の人には関係ない”といった短絡的な思考に陥りかねない。日本語訳が悪いために意味が限定されてしまった一例だ。

私にとってはスプレッドシートもゲームも同じX68000上のアプリケーションにすぎない。

1. 銀行と小金とせこい利息かせぎ

などといいつつ，実にスプレッドシートらしいお金の計算なんてしてみようかと思う。真剣にやろうと思うと小数点以下の処理や大きな値を扱うときの誤差なんかも無視はできないが，私は銀行屋ではないのでそこまでは考えない。

老若男女問わず，我々は銀行なり郵便局なりにお金を代表とする資産（大袈裟な言葉だこと！）を預けている。貸しているといってもいい。銀行はその集めた金をまたほかのところへ貸して高い利子を取り，我々にはそこから手数料を引いた安い利息をつけてくれるわけである。

で，銀行といっても日銀から第一勧銀から街の農協までいろいろある。まず日銀(日本銀行)。こいつは銀行の総元締めで，全国の銀行は日銀から金を借りたり，日銀に金を預けたりしている。日銀はおいておいて，その他，銀行には以下の種類がある。都市銀行，地方銀行，相互銀行，長期信用銀行，信託銀行，信用金庫，信用共同組合，労働金庫，農協という感じだ。いや，べつに銀行講座をやるつもりはない。

そのほか，郵便局と証券会社も忘れてはいけないな。あと，地方銀行と相互銀行から鞍替えした第二地銀を分けるのもいいかもしれない。

でもって，お金がとびかっているわけだが，たいていの場合，磁気ディスク上のデータが行ったり来たりしているに過ぎない。しかも大きな金額を動かしているのは企業が中心であるから我々庶民にはあまり縁のない話ではあるな。

今回は「預けて増やさせる」ということを中心に考えて計算をしてみようと思う。我々が預金できるのはたいした金額ではないので，使える預金の種類も少ない。また，我々庶民の手の届くような預金の金利（つまり，利率)っていうのは規制金利といって利率が決まっているので，どの銀行だと得だということもない。

まず，固定金利にするか変動金利にするかだ。が，何がお得かという話などする気は毛頭ないし，ましてやこれはOh!Xである。間違ってもその辺の週刊誌のマネー講座と一緒にしないように。

さて，固定金利っていうとお金を預けた時点の金利が満期日まで変わらないものである。普通銀行が扱う預金は基本的に固定金利だ。普通預金ってのは例外。

変動金利というのは金利の変動がそのとき預けていたお金にそのまま反映するものである。たいてい半年複利っていう計算をしているので，半年ごとに利息がでる。そのとき，次の半年間の金利（利率）にその時点での金利が反映するのだ。信託銀行が扱っている商品（商品ってのもピンとこないが）が基本的に変動金利だ。

●金利と利息と税金と金融商品

具体的にいこう。

善良な市民は銀行に“普通預金”の口座というのを持っていて，そこに給料が振り込まれたり，そこから丸井やセゾンのクレジットを払ったり，株の下がったNTTやら原発の好きな東京電力やらが定期的に金を引き出していく。まあ，財布みたいなものだ。だから，利率も微々たるもので，雀の涙みたいで，銀行のほうも普通預金が多くてもあまりおいしくない。

で，普通預金にある程度余裕ができてくると，もっと利率がいい手段に訴えたくなってくる。銀行の女の人もやたらいろいろと勧める。代表的なのが普通銀行の場合，定期預金（期日指定定期）か定期積み立て預金だ。これが100万円以上あるなら，小口MMCっていう新しい技もある。

ここで貪欲な人は普通銀行以外にも手口があることを知る。信託銀行へ行けば変動金利で一見お得なヒット(金銭信託の一種)やビッグ（貸付信託の一種）が，証券会社へ行けば中国ファンドがあるわけだ。こんなところがポピュラーで，元金保証がなくてかえって損をする覚悟があるなら投資信

託っていうのもある。

そこで４年前。昭和61年の６月になけなしのボーナスから20万円ほど捻出して２年間預けたとしよう。どうして４年前なのかはおいおいわかることである。

選択肢は先に挙げたようにいろいろあるが，期間を考えて，普通銀行でポピュラーな期日指定定期，信託銀行でポピュラーなビッグとしてみた。20万円を２年間預けるというとなると，このどちらかが妥当なところだ。その利率を見てみよう。

当時，２年ものの定期預金の利率は年4.38％だった。期日指定定期ってやつで２年預けると１年複利で計算ということになるので，そうする。複利というのは，利息が出るたびに，元金にその利息を足したものに対して次の利息が計算されるというもの。

２年ものの貸付信託の予想配当率は年4.58％だった。予想配当率というのは，絶対その利率で利息が返せるとは限らないよ，という意味なのだが，実質的に利率と考えて間違いないそうである。

ビッグってのは半年複利型の貸付信託である。信託ってのはまあ，信託銀行が扱うことのできるもので，変動金利の金融商品だと思っておけばいいだろう。詳しく話すと長くなるからやめておく。

さあ，Kamikazeである。

まず，期日指定定期だ。これは固定金利のために計算は簡単。

20万円×1.0438^2

(元金×(1＋利率/100)^年数)

である。いちおうスプレッドシートらしく預金額と利率は別のセルに入れておいた。なお，この式は１年複利である。２乗しているのは利息計算が２年で２回出る複利計算だからである。

続いてビッグだ。これは変動金利である。ビッグは半年複利。変動金利の半年複利。これは半年ごとに利息を出して利率を見直す。計算するときは半年ごとに(利率＋1)を掛けていってやればいいわけだ。年利回りが４％なら半年当たり２％となる。計算式は，

元金×(1＋最初の利率/2/100)×(1＋次の利率/2/100)×……

である。これが図１の解説だ。

昭和61年といえば，公定歩合が低く低くなっていった時期であり，それに影響を受けて金利も安かった（金利が低く景気がよくなってきたために，株があれだけはやったのだともいえる）。そのあたりの事情は図２のグラフにある。Kamikazeのグラフ作成の能力を越えたものになってしまった（横軸の目盛りが細かいため，うまく表示されなかったようだ）ので，少々見苦しい。

公定歩合ってのは，日銀が民間の銀行に対してお金を貸し出すときの利率である。こいつの上下が金利や景気に影響を及ぼす。歩合ってのは割合のことで，公ってのは御上のことだから，御上の定めた割合って意味だと思えばいい。景気をよくしたいときは下げ（安い金利で金を借りられるので，金を借りてことを起こす連中が増える），景気がいいときは上げるのが普通らしい。

で，結果として変動金利のビッグが損をしている。これは珍しいケースで，定期預金より利率の高いビッグでもこういうことはあるのだ。

しかし，公定歩合が上がり始める63年12月に預けたとしたらどうだろう。

というわけで，結果が図３である。図１とあわせて見れば銀行に金を預けるということがどういうことか少しはわかるだろう。

あと，昭和63年から導入されたマル優廃止による“利息の20％は国のもの”という乱暴な政策により，結果として受け取れる金は減ったりする。ああ，前門のマル優廃止，後門の消費税というわけで，このときより貯めても使っても税金がかかるようになったのだ。いくらなんでも２割は多いが，閑話休題。

ついでだから，平成２年10月から２年間預ける場合，金利は変わらないとして計算したのが図４である。

新しく飛び込んできた中国ファンド（中期国債ファンドといって，証券会社はその30％以上を中期国債で運用することに法律で決められている）は１カ月複利であるから，結果としてお得なように見えるが，税引き後を見ると，ビッグより安い。

これは，中国ファンドが１カ月ごとに20％の税金を取られるのに対し，ビッグは満期時の利息に対して税金がかかるようになっているためである。まあ，満期まで取り出せない（中途解約すると手数料を取られる）ビッグに比べ，いつでも引き出せる中国ファンドの魅力はある。

利息にかかる税金の計算であるが，ビッグのように満期時に一気にかかる場合は，満期時に受け取るはずの金額から元金を引いて（これが利息ね）それに0.8を掛けて

図1

昭和６１年６月末の場合

	定期預金２年間の場合	
	元金	¥200,000
	利回り	4.38%
	預入期間	2
	満期時	¥217,904

	ビッグ２年間の場合	
	元金	¥200,000
	預入期間	2
利回り	61/06	4.58%
	61/12	4.21%
	62/06	3.84%
	62/12	3.84%
	満期時	¥216,985

図3

昭和６３年１２月末の場合

	定期預金２年間の場合	
	元金	¥200,000
	利回り	3.64%
	預入期間	2
	満期時	¥214,825
	税引後	¥211,860

	ビッグ２年間の場合	
	元金	¥200,000
	預入期間	2
利回り	63/12	3.84%
	1/06	4.40%
	1/12	4.77%
	2/06	6.08%
	満期時	¥219,777
	税引後	¥215,822

図2

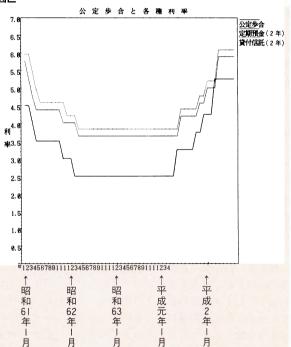



図4

平成２年１０月１５日現在の利率による計算

元金　¥200,000

期間　2

	利率	２年後	税引後
期日指定定期（２年	6.33%	¥226,121	¥220,897
ビッグ（２年）	6.53%	¥227,427	¥221,942
中国ファンド	6.497%	¥227,672	¥221,860

やれば税引き後の利息はわかる。中国ファンドのようにその都度税金を取られる場合は，あらかじめ利率に0.8を掛けて計算すればいい。

利率の差なんて2年程度ではそんなに出るものではないね。これが5年くらいになると差は開いてくるけれど。まあ，計算してみると面白いだろう。5年ものの貸付信託の利率はいまのところ，年8.02%である。一般に1年複利より半年複利が，それより1カ月複利がお得なことになっている。

今のご時世，明日は明日の風が吹くので，何が得かといわれても，困るが，少なくともKamikazeでも使って計算してみれば，一見何が得かわからないような数字に関しても具体的に判断できていいこともある。

ちなみに，非常にお得な大口定期という自由金利商品もあるのだが，これは最低預け入れ額が1000万円なので，考えもしなかった。いつかは大口定期を使える身分になってみたいものである。

ちなみに，こういった利率は新聞（朝日新聞なら週末の夕刊）に載っている。

あと，預ける話ばかりしたが，金利が高いということは，預けるときにも借りるときにもいえることなので，ローンを組む場合は不利である。金利の安いときに固定金利でローンを組んだ人は，結構お得感があるだろう。Kamikazeの財務関数を使えば借りた金を返済する計画を立てる助けにもなるが，とりあえずそういう縁起のよくなさそうな話は避けておこう。

2. 売上管理のおもちゃ的例題

スプレッドシートというと，日常業務でデータ処理に使うか，ある程度まとまった表を入力してその分析に使うかどちらかというのが一般的のようだ。

意志決定支援（なんて大袈裟な言いぐさ！）に使うのもいいらしい。ひと通りデータを入力したあと，製品を5%引きにして売り上げが10%伸びたとしたら果たしてどのくらいメリットがあるか，ってな計算をさせるのである。

まあ，とりあえずそういった使い方に焦点を当ててみる。商品Aなどとやってもつまらないので，今発売されているX68000を表にしてみた（図5）。面倒なので，本体のみである。この表に製品コードという名前をつけてセーブしておく。

さて，売上管理である。記録するときはなるべく楽をしたい。いちいち商品名を打ち込むより，その製品に付けられた製品コードと数量を打ち込むだけであとはパソコンがよきにはからってくれるというのが望ましい。

そこで，図6である。たとえば，こんな結果になる。一見なんということはないが，実際に打ち込んだのは製品コードと数量だけだ，というのがミソ。

ここではVLOOKUP関数を使ってみた。

＝VLOOKUP(B3,製品コード！A3：C10,1)

っていうのは，図5の“製品コード”という名前のシートのA3からC10を対象にして，B3セル（売上表の製品コードが入っているセル）と同じものを探し，その右のセルの値（製品名）を返すというものだ。厳密には同じものを探すわけではなく，値を比較しているだけなので，製品コード表は製品コードで昇順にソートされている必要がある。で，製品コードのシートではA列に製品コードが，B列に商品名が，C列に単価が入っているので，A列のひとつ右は製品名，2つ右は単価になるわけだ。

ここでは製品コードという名前でセーブした図5のシートを製品コードテーブルとしてアクセスしている。そのためには環境設定のパスにそのファイルがなければならない。この関数に限らず，Kamikaze君は他シートの参照時にディスクを読みにいくので少々時間がかかると同時に，ディスクに最新版をセーブしておく必要がある。

この“他シート参照技”を覚えると，けっこう応用が利くので便利である。普通，データベースなんかを使うとき，いくつかのパターンしかない項目についてはそのコードを設定しておき，コードを入れるだけで済ますのが基本だ。

たとえば，地区コードを47都道府県に割り振っておけば，入力時に，数字を打ち込むだけで県名が出てくるようにできる。いちいち変換するより，数字を打ち込むほうが早い。

蔵書管理の場合，その本の大きさによってコード001は文庫で，002は新書でってやっておくと，入力時に数字を入れるだけで，あとはVLOOKUP関数で“文庫”っていう文字が自動的に入るようにできるのだ。なんか，コンピュータを使っている気分である。

さて，先月，Kamikazeは大きなシートを作ると遅さが際立ってしまって損をした気分になるという話をした。よって，他シート参照を多用すれば，1つひとつの表が小さくても十分活用できるのである。

たとえば，1カ月分の日ごとの明細シートを作り，1カ月分集計した結果のシートを作ってそこに集計し，12カ月分集まったら，年間集計シートを作ってそこに集計するという作業をすることも可能だ。

もちろん，自動再計算はOFFにしておく。いちいちディスクにいくのはとてもうっとうしいので，再計算は手動だ。

3. 集計作業も関数でできてしまう

で，図7である。ちょっと趣向を変えて，X68000の毎日の売り上げを何割引きで売ったかも加味して作ったものである。あくまでもサンプルね。紙面の都合もあって，6日分だけである。ここから，データ範囲関数ってのを使って，機種ごとの合計を算出した(図8)。データ範囲関数というのは，任意の範囲のデータに対して条件にマッチしたものだけを対象に行う統計関数である。しごく便利だが，しごく面倒臭い。

たとえば，X68000EXPERTII（ブラック）つまりCZ-603C-BKの合計販売数を求めるのは，

＝DSUM(元データ,3,G18：G19)

である。元データっていうのは図7の範囲につけた名前だ。3ってのは，元データの3列目に売った個数が入っているからだ。で，G18とG19にはマッチングの条件が入っている。

G18にはマッチング対象が2列目の製品コードだから2。

G19はCZ-603C-BKを検索するのだから，

_＝"CZ-603C-BK"

図5

製品コード表

製品コード	商品名	単価
CZ-603C-BK	X68000 EXPERTII	¥338,000
CZ-603C-GY	X68000 EXPERTII	¥338,000
CZ-613C-BK	X68000 EXPERTII-HD	¥448,000
CZ-623C-TN	X68000 SUPER-HD	¥498,000
CZ-653C-BK	X68000 PROII	¥285,000
CZ-653C-GY	X68000 PROII	¥285,000
CZ-663C-BK	X68000 PROII-HD	¥395,000
CZ-663C-GY	X68000 PROII-HD	¥395,000

A3←（CZ-603C-BK）　→C10（¥395,000）

図6

10月X68000売上表

製品	数量	製品名	単価	合計
CZ-603C-BK	23	X68000 EXPERTII	¥338,000	¥7,774,000
CZ-613C-BK	19	X68000 EXPERTII-HD	¥448,000	¥8,512,000
CZ-653C-BK	10	X68000 PROII	¥285,000	¥2,850,000
CZ-653C-GY	18	X68000 PROII	¥285,000	¥5,130,000
CZ-623C-TN	19	X68000 SUPER-HD	¥498,000	¥9,462,000
CZ-663C-GY	8	X68000 PROII-HD	¥395,000	¥3,160,000
合計	97		¥2,249,000	¥36,888,000

ここだけ入力　　こっちは自動

図7

10月X68000売上明細

日付	製品コード	数量	製品名	割引率	売値	合計
1	CZ-603C-BK	3	X68000 EXPERTII	10%	¥304,200	¥912,600
	CZ-603C-BK	1	X68000 EXPERTII	20%	¥270,400	¥270,400
	CZ-613C-BK	2	X68000 EXPERTII-HD	25%	¥336,000	¥672,000
	CZ-623C-TN	1	X68000 SUPER-HD	18%	¥408,360	¥408,360
	CZ-663C-GY	1	X68000 PROII-HD	30%	¥276,500	¥276,500
2	CZ-653C-GY	2	X68000 PROII	20%	¥228,000	¥456,000
3	CZ-613C-BK	3	X68000 EXPERTII-HD	10%	¥403,200	¥1,209,600
	CZ-653C-GY	2	X68000 PROII	25%	¥213,750	¥427,500
4	CZ-623C-TN	2	X68000 SUPER-HD	32%	¥338,640	¥677,280
	CZ-653C-GY	1	X68000 PROII	25%	¥213,750	¥213,750
	CZ-653C-GY	2	X68000 PROII	20%	¥228,000	¥456,000
5	CZ-613C-BK	3	X68000 EXPERTII-HD	15%	¥380,800	¥1,142,400
6	CZ-603C-BK	5	X68000 EXPERTII	30%	¥236,600	¥1,183,000

元データ

図8

10月X68000集計

	数量	売り上げ合計
CZ-603C-BK	9	¥2,366,000
CZ-603C-GY	0	¥0
CZ-613C-BK	8	¥3,024,000
CZ-623C-TN	3	¥1,085,640
CZ-653C-BK	0	¥0
CZ-653C-GY	7	¥1,553,250
CZ-663C-BK	0	¥0
CZ-663C-GY	1	¥276,500
合計	28	¥8,305,390

条件1	2	2	2	
条件2	_="CZ-603C-BK"	_="CZ-603C-GY"	_="CZ-613C-BK"	_="CZ-62:
条件1	2	2	2	
条件2	_="CZ-653C-BK"	_="CZ-653C-GY"	_="CZ-663C-BK"	_="CZ-66:

DSUM関数用条件式

である。つまり，元データから製品コードがCZ-603C-BKであるものを引っ張り出して，その販売数を合計しなさい，てな指令になるのだ。なんて便利。スプレッドシートらしい使用法の一例だ。

ただし，これはこういう小さい表だからいいのであって，1日に何十項目も入力のある業務で使って実用になる速度を得られるかどうかの保証はしない。

4. 初歩的な数学とスプレッドシート

風の便りに聞くと，本場アメリカでは科学技術計算などにもスプレッドシートは使われているそうである。なんでも，循環参照機能を使って，初期値を指定した再帰っぽい計算のできるものもあるそうである。

Kamikazeでも簡単な数学系の関数は用意されている。循環参照計算もできる。

循環参照とは，A1セルに＝A2+5とあって，A2セルに＝A1+2とある場合，A1を計算するにはA2セルの値が必要で，A2セルを計算するにはA1セルが必要という互いに参照しあうものだ。デフォルトでは“こういうことをされては困ります”ということになっているが，解禁することも可能だ。

解禁するとどうなるかというと，無限に参照しあってしまう。いつまでも計算してオーバーフローで止まるか，本当に止まらない。それを防ぐために，反復回数の指定ができるようになっている。

ここで私は思った。うまく収束してくれる関数であれば，簡単に結果が求められるのではないか。が，そうは問屋が卸さない。初期値の指定ができないので，掛け算や割算が入るとうまくいかないのだ。よって，階乗の計算なんかも無理だった。残念。

それでは，ということで，連立方程式に挑戦してみた。図9である。

まずセルを2つ使い，式を変形してX＝の式とY＝の式にする。それぞれをセルに割り当て，循環参照させる。解に向かって収束していってくれるはずなので，循環回数を増やすほど（Kamikazeの扱える有効数字を限度として）答えが求められるはずだ。

扱った式は，次の3つだ。

Y＝X^2，Y＝X＋2
Y＝X^2，Y＝2X＋2
Y＝3X＋20，Y＝9X

一見うまくいっている。しかし，なんか変だ。連立2次方程式の解は2つあるのだが，図ではひとつしか求まっていない。どこに問題があったのかは一目瞭然。最初の解に向かって収束していくだけなのだ。

具体的には，Xが正である最初の解に向かって収束する。釣り合ったら，そこで同じ値のまま無駄に計算を続ける。負の領域にある解を無理やり求めようとしたら，値が発散して（X＝30000，Y＝－30000てな感じ）しまった。

連立1次方程式の場合，素直に答えが出ているように見えるが，それも答えが出るような式を作ったからであって，解が負になったりすると，収束しないで発散してしまう。考えればすぐにわかる。まあ，循環参照のサンプルとでも思ってくれたまえ。

図9では循環を10回しかさせてないが，図10では40回させたものを載せた。収束するのである。ちなみに，Y＝X^2とY＝X＋2のXとYが収束していく様子だ。

誰か，いい循環参照計算の使い道を思いついた人は教えてください。

最後はおまけ。三角関数をπを10等分して計算し，グラフにしてみた。sin，cos，θ－sinθ，sin×cosの4つである。

式の視覚化はKamikazeの得意とするところなので，中学数学から高校数学の簡単なところなら，結構楽しく遊べる。

＊　＊　＊

というわけで，Kamikazeシリーズも2回目が終わった。来月はKamikaze Ver.2でやっと採用されたプログラム機能やデータベース機能の話をするつもりがある（あくまでもつもりであるが）。なるべくKamikazeを持っていなくても読めるようにはする予定だが，そうならなかったら，ごめんなさい。

参考文献

［1］金利・利回りがわかる事典　角川総一著　明日香出版社刊
［2］信託の知識　川崎誠一著　日本経済新聞社刊

図9

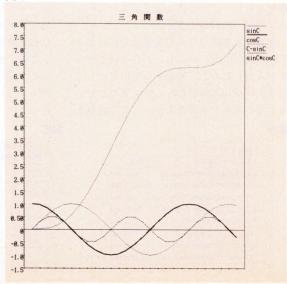
連立方程式を解く（反復回数10回）

2次方程式	X= 1.9999976469	X=SQRT(Y)
	Y= 3.9999905876	Y=X+2
	X= 2.7318679265	X=SQRT(Y)
	Y= 7.4631023677	Y=2X+2
1次方程式	Y= 29.999491947	Y=3X+20
	X= 3.3331639825	X=Y/9

図10

連立方程式を解く（反復回数40回）

2次方程式	X= 2	X=SQRT(Y)
	Y= 4	Y=X+2
	X= 2.7320508076	X=SQRT(Y)
	Y= 7.4641016151	Y=2X+2
1次方程式	Y= 30	Y=3X+20
	X= 3.3333333333	X=Y/9

図11

コンピュータシミュレーションの世界

Kamon Masato 華門 真人

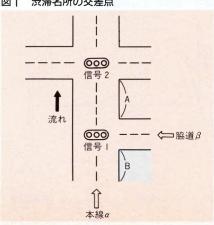
現象をモデル化して状況を分析し新しい状況を予測する。これはもっともコンピュータらしい仕事ではないでしょうか。この連載ではシミュレーションの基本的な考え方から制御言語の作成まで，集中的に対応していく予定です。

すべての始まりはクルマに乗っていたときのことである。クルマっていうのは，かの有名な（？）アルシオーネのこと。ここしばらく顔を見せないと思ったら，相変わらずクルマを乗りまわしている筆者なのでありました。

僕がクルマに乗るようになってからもう３年以上になる。この３年のうちに，ベイブリッジが開通したし，首都高速神奈川線の狩場線も（ローカルな話題でゴメン）開通した。道路自体は次第に整いつつあるように見える。

しかし，しかしだ。これはあくまで個人的な感想なのだけれども，渋滞は年々ひどくなっていっているような気がする。確かに道路の整備によって交通事情が改善されたケースも，少ないながら，ある。完全とはいえないけれども，横浜新道がそのいい例だろう。

が，その一方で一般道路での渋滞は年々激化しつつある。特に交差点の近辺がひどい。最近目立ってひどいのが日比谷通りの芝公園付近だ。あ，またまたローカルな話題で申しわけない。でも編集室も芝公園とはご近所の泉岳寺になったことですし，許してくださいませ。

交通事情が悪化してきているというのは最近いわれ始めたことではないのだが，個人的には最近特にひどいような気がする。なぜなんだろう。直接的な原因としては車両の増加があるのだろう。なんたって最近のクルマの売れっぷりたるやすさまじいものがある。

でもそれと同時に，もうひとつの間接的な理由があるような気がする。表現が難しいのだけれど，「交通無策」とでもいおうか。要するに，交通をうまく流すための十分な対策が立てられていない，ということ。

具体的にいうと，車線規制，信号の可変制御なんかのことだ。確かにこれらの対策は立てられつつはあるのだけれど，十分でない，というところがほとんどなのだ。首都高速の箱崎なんかが，特に有名だよね（TVでもとりあげられているから知っている人も多いだろう）。

そろそろ本題に戻らないと，話がどこかに行ってしまいそう。要するに僕がクルマの中でなにを考えたのか。僕はこう思ったのだ。渋滞をコンピュータでシミュレートすることはできないのだろうか。そしてうまくいけば，渋滞解消法を見つけられないだろうか，と。

こうして僕は，コンピュータシミュレーションの世界への第一歩を踏み出した。

図１　渋滞名所の交差点

WHAT

まず僕が考えてみたのは，交差点の様子をうまくシミュレートできないだろうかということだ。これには理由がある。クルマに乗っていてよく通る交差点があるのだが，日中はいつも渋滞している。

よく観察した結果，これは信号のタイミングが悪いのではないか，と考えるようになった。図１のような交差点なのだが，脇道βが合流してくるところで非常に混雑する。

素人考えで悪いのだが，自分なりに渋滞の仕組みを推測してみた。まず信号１が赤になって本線αの流れが止まる。逆に脇道βからは図のＡの部分にクルマが流れ込んでくる。ここで信号２も赤になると，図のＡの部分にクルマが滞り，すぐにＡはクルマで一杯になってしまう。

しばらくのちにまず信号１が青になる。が，Ｂにいるクルマは Ａが一杯になっているために動けない。信号２も青になり，ようやくＡにいるクルマが流れ始めてＢが進もうとすると，ちょうど信号１が赤に変わってしまう。

このようにして，脇道からのクルマが比較的コンスタントに流れるのに対し，本線はほとんど動けない。その結果，激しい渋滞となる，というわけだ。

もちろん，この渋滞の第一の原因はクルマが多すぎることにあることは確かだ。その証拠に，クルマの台数が少なくなる夜間はこの交差点もスムーズに流れている。しかし，それは十分に理解したうえでも，この信号によっても渋滞が加速されているような気がしてならない。

所詮は素人のあさはかさなのかもしれない。実はこの信号の動作には，素人には理解しえない奥深い理由があるのかもしれない。確かに，知識だけからいえば，交通制御の専門家にはかなわないと思う。でも，我々には強い味方がいる。そう，コンピュータだ。この交差点の様子をコンピュータの中に再現できたら……。

うまくシミュレートしてやれば，この素人考えを実証することもできる。さらに，シミュレーションの結果を利用して，信号をうまくコントロールしてやれば，あわよくば渋滞を緩和させることができるかもしれない。

HOW

さて，渋滞を解消させるなどと目標は高く掲げてみたものの，どうやってシミュレートしてやればよいのだろう。

まず最初に考えてみたのは，クルマ１台

１台の動きを精密にシミュレートできないだろうか，ということである。１台１台がシミュレートできれば，当然その集合体である交差点もシミュレートできるはずだ。

クルマの動きをよく見ているとわかると思うが（もちろんクルマを運転している人ならなおさらわかると思うが），クルマの速度はだいたい先行車の速度と，先行車との車間距離によって決まる。え，先行車がいない場合は，って。そんなときは先行車がかなり遠くを標準的な速度（60km/hぐらい）で走っていると考えてやればいい。もっとも渋滞シミュレーションじゃあ，なかなかそんな状況は生じないだろうけれど。

ま，ともかくこのように考えてみると，交差点に入ってくるクルマ１台１台に対し，先行車との関係（すなわち，先行車の速度と先行車との車間距離）から速度を割り出してシミュレートする，というモデルが考えられる。

このモデルの命となるのが，先行車との関係から自車の速度を割り出すという部分。クルマが走っている様子を頭に思い浮かべてみよう。前のクルマがどんどん離れていったらどうするか。まあ，普通は加速して追いつこうとするでしょう。

これは要するに，

IF 車間距離＝増加 THEN 加速

のように考えることができる。同様にして，

IF 車間距離＝一定 THEN 速度維持
IF 車間距離＝減少 THEN 減速

であるかのように思える。

しかし，これでは間違い。なぜなら，車間距離がずいぶん開いてしまった場合には，「加速」して車間距離を「減少」させようとするだろう。

これはむしろ，先行車との適正な車間距離というものを設定し，実際の車間距離が適正値より大きかったら加速，などと考えたほうがよさそうである。

つまり，

IF 車間距離 > 適正値 THEN 加速
IF 車間距離 ＝ 適正値 THEN 定速
IF 車間距離 < 適正値 THEN 減速

と考えることができる。

さて，これで骨組みはできた，さあプログラムを書こうというのは少し気が早い。まず車間距離の適正値はどうするか。これは当然車速に応じて変わってくるだろう。高速のときは通常，車間距離を多めに取るものだけれど，低速時は車間距離は少なめになってくるだろう。さらに完全に止まってしまった場合は，だいたい１mぐらいが適正値だろう。

ということは，適正値は車速の関数として書けることになる。すなわち，

適正値＝ｆ(車速)

というわけ。

まだ問題はある。加速するといっても，どのように加速するのだろう。どれぐらい車間距離が変動しているのかによっても加速の度合いは変わってくるはずだ。具体的にいえば，車間距離が大幅に適正値を上回っていたら派手に加速するだろうし，少しだけだったら徐々に加速するだろう，ということ。

さらに加速の際には最初は加速度が大きく，徐々に加速度が小さくなっていく，ということも考えねばならない。

以上を総合してみると，加速の大きさは車間距離の関数になる。さらに加速の際の加速度の変動のしかたは，独自の関数に従うことになる。そしてその加速度を積分してやれば，求めたい速度が，ようやく，得られる（図２を参照してほしい)。

こうして見てみると，このシミュレーションは関数の嵐になることがわかる。まあシミュレーションなんていうのはもともと演算の積み重ねでできるものだから，当たり前といえば当たり前なのだけれども。

ところが，実際にプログラムを書いてみたところ，大パニックに陥ってしまった。とにかく複雑になってしまい処理しきれなかったのである。いろいろ簡素化して実現しようとしてみたのだが速度的にも辛いものがある。

下手をすると実際の交差点モデルよりも遅くなってしまいそうだったのである。もともとシミュレーションは将来をより簡単に，より早く知るために役立つはずのものなのであるから，これでは意味がない。

結局，あろうことか僕は匙を投げ出してしまったのである。しかしもちろん，努力次第ではこのやり方でうまくいくはずである(誰かやってみてください)。しかし，面倒臭い！

さて，こうしていきなり試みは挫折してしまったのである。この連載の運命やいかに……。

STEP BACK

これはなんについてでもいえることだけれども，行き詰まってしまった場合はどうすべきだろうか。

まあ，普通は一歩下がってもう一度よく考え直してみる，といったところだろう。そうすることによって新しい方法が見えてくることもある。例にもれず，ここでももう一度いろいろと考えてみることにする。

物事をシミュレートする場合，現実をどのように模倣するか，というのが大きなポイントになる。いい換えればどのようなモデルにするか，ということである。

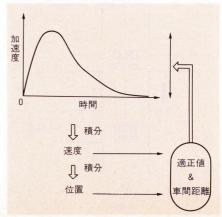
ひと口にモデルといっても，２種類のモデルを考えることができる。ひとつは「物理的モデル」という代物である。この代表例が大きさを縮小するスケールモデルである。交差点のシミュレーションの例でいえば，10分の１の大きさのクルマや信号などを作って動かすことによりシミュレートするモデル，ということになる。

もうひとつは「数値モデル」である。これは現実を数値に置き換えることによって構成される抽象的なモデルのことで,「数値を扱うことならおまかせ」というコンピュータでシミュレーションをするとなると，当然数値モデルを考えることになる。

ただ，数値モデルといっても１種類とは限らない。現実の要素をどのようにとらえるかで異なるモデルができてくる。その要素のなかでもっとも重要なものが「時間」であり，その「時間」をどのようにとらえるかで２種類のモデルができあがる。

違いは時間を連続するものと考えるか，あるいはポイント，ポイントがつながったものと考えるのかという点にあり，前者は「連続変化モデル」，後者は「離散変化モデル」と呼ばれている。

連続変化モデルは，時間を連続的に（もちろんコンピュータ上では微小な等時間間隔で）変化する値であると考え，時間を中心にしてモデルを変化させていく。逆に離散変化モデルでは時間というものは，モデルが変化する重要なポイントをつなげたものとして表現される。いい換えればモデルの変化に応じて時間が（不等時間間隔で）刻まれていくということになる。

図2　加速度算出の概念図

複雑になってきたので具体例で考えよう。たとえば右折レーンのある交差点だ。連続変化モデルは時間を中心にモデルが変化していく。すなわち右折車が右折レーンに入ってそのまま右折レーンを走っていき，交差点の真ん中で一旦停止して，対抗車線が途切れたのを見計らって発進し，対抗車線を横切って右折し終わるまでのあいだ，ずっと時間は連続的に流れていく。もちろんコンピュータでは完全に連続な時間というものは表現できない。そこで実際には0.1秒ごとなど，微小な時間ごとに時間を刻む。0秒，0.1秒，0.2秒……などのように時間をはかり，時間に応じてモデルを連続的に変化させていく。

これに対して離散変化モデルではどうだろうか。離散変化モデルではモデルの変化によって時間が刻まれていく。上の例でいえばモデルが大きく変化するポイントは，右折車が右折レーンに入る，交差点の真ん中で停止する，再発進する，右折し終わるという4点であるから，この4点のみに対し，時間が刻まれていく。逆にいえばこれら4つのポイントのあいだは無視されてしまい，これら4つ（だけ）をつないで時間を離散的に表現してしまう。

あえて簡単にいってしまうと,「いま時間がこうだから，モデルはこうなる」というのが連続変化モデル,「モデルがこうなったから，時間はこうなる」というのが離散変化モデルといえる。

賢明な読者諸兄はもうお気づきだと思うが，前章でトライした方法は典型的な連続変化モデルである。時間を中心におき，速度などの状態を時間の関数として連続的に表現しようとしているのだから。

ここまでくればもうおわかりでしょう。連続変化モデルに挫折してしまった以上，ここは離散変化モデルにアタックするしかない！　わけだ。

HOW Part2

おそらく多くの人はまだ，離散変化モデルには馴染んでいないのではないかと思う。なんとなくわかりにくいことは確かだ。

連続変化モデルのほうは比較的理解しやすいだろう。なんといっても現実では時間は連続なのだから。なにかを時間の経過とともに見ていけば，それで連続変化モデルになる。

それに比べて離散変化モデルはどうもしっくりこないかもしれない。あるいはそんなので交差点のシミュレートなんかできるのだろうか，と考える人もいるだろう。しかし，さすが時間をぶつ切りにしてしまうだけあって，離散変化モデルというのはモデルが単純で，一旦理解してしまえばいろいろと応用もきく。

ここでは，遠回りにはなるかもしれないが，離散変化モデルに取り組んでみることにする。もちろん最初は単純なモデルから始め，やがて交差点などの複雑なモデルにたどりつけたらお慰み，である。

それでは一番簡単な例として，図3のようなモデルを考えてみることにしよう。

*

最近，一部の有料道路で通行券が機械によって発券されるところがある，ということをご存じだろうか。

念のため解説しておこう。有料道路で料金を徴収する場合，2通りの方法がある。ひとつ目は道路の途中に大きな料金所をもうけ，そこで一斉に徴収する方法。第3京浜道路や，多少変則的だが首都高速道路がこれにあたる。この方法は主に料金が均一な場合に用いられている（首都高速道路は600円均一，ただしいまのところは，であるが)。もうひとつは，よりメジャーな方法だが，入り口（IC，インターチェンジ）でどのICから乗ったかを証明する「通行券」を渡し，出口のICでそれをもとに料金の精算をする方法。

東名自動車道路を始めとするいわゆる「高速道路」はすべてこの方式だし，私事で恐縮だがよく利用している横浜横須賀道路もこの方式である。この方式は一般的に，料金が均一でない場合，すなわち料金が走行距離に応じて異なってくる場合に用いられている。

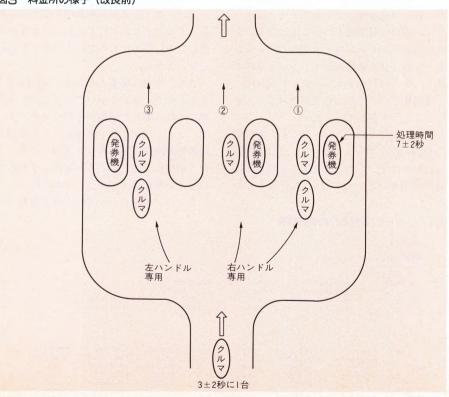
ここで取り上げたいのは後者である。最近，横浜横須賀道路の一部のICで通行券の発券が機械によって自動化された。いままではおじさんが手渡してくれたものを，機械がクルマを感知して自動的に発券するようになったのである。

もっとも，正確にいえば，そういうものらしいということしか知らない。ただ噂によると，僕自身がよく利用するICも近々自動化されるらしいという。

ここで問題となってくるのが左ハンドルのクルマの存在である。いままではドライバーも発券するおじさんも手を伸ばしてなんとかなっていた。ところが相手が機械ともなるとそうもいかない。かといって左ハンドルのドライバーにはクルマを降りてもらうというわけにもいかない（それが原因で渋滞になってしまう)。

実際にはどうしているのかというと，実は左ハンドル専用の発券機なるものが設置されているらしい。ここまでを理解したうえで図3のようなモデルを考えてみよう。

図3はある仮想の料金所である。料金所とはいっても入り口のICであるから，仕事

図3　料金所の様子（改良前）

は通行券の発券のみ。発券用の車線は３車線あって，そのうち２つ（第１レーン，第２レーン）が右ハンドル専用，すなわち右側に発券機があり，残りのひとつ（第３レーン）が左ハンドル専用，すなわち左側に発券機が設置されている。

それではこのモデルをシミュレートしてみることにしよう。シミュレーションの前提として，発券にはどの発券機でも５～９秒（７±２秒）のあいだの時間（整数秒）を等確率でとり，クルマは混雑時らしく１～５秒（３±２秒）間隔でやってくるとする。なお，クルマ全体に占める左ハンドルの割合は10％であるとする。

ICに入ってくるクルマは左ハンドルなら無条件に第３レーンに進む。右ハンドルの場合，第１レーンか第２レーンのうち，すいているほうに進む。もし同じように混んでいる場合は等確率で第１レーンか第２レーンのどちらかに進む。

以上の条件に従ってBASICでプログラムを書いてみたのがリスト１である。今回は，どのようにしてプログラムを組んでいくかということよりも，シミュレーションがどのように有効かを中心に見ていこうと思う。そこでさっそく表１がリスト１の実行結果である。

おっとその前に新しい言葉を３つ覚えてもらおう。その３つとはトランザクション（Transaction，略してXact），ファシリティ（Facility），キュー（Queue）のこと。順番に説明していこう。

トランザクションとは「時間の経過とともにシステムを動いていく対象」をモデル化したもののこと。というと難しそうだが，シミュレーションのシステムの中を動いていくものだから，システムの「コマ」とでもいおうか。この例の場合はシミュレーションのシステム（IC全体）の中を移動していくもの，すなわち，クルマがトランザクションということになる。

システムの中を動いていくトランザクションに対し，ファシリティは動かない対象をモデル化したものである。定義は「同時に単一のトランザクションのみが使用しうる機器設備類」ということになる。

動いてくるトランザクションを処理していくのがファシリティということになるが，同時にひとつのトランザクションしか利用できない，というところがミソである。トランザクションをシステムの「コマ」にたとえるとすると，ファシリティは「マス」とでもいおうか。

この例でいえばシステム（IC）の中にあって，トランザクション（クルマ）が利用する対象であるから，「発券機」ということになる。発券機は，もちろん，移動しないし，同時に１台のクルマしか利用できない。

さて，発券機を利用するのはいいのだが，必ずしもすぐ利用できるとは限らない。ときには混雑しているため列に並んで待たなければならないこともあるだろう。このとき生ずる「待ち行列」のことを指してキューと呼ぶ。

以上をまとめると，「システムの中をトランザクションが移動していく。その途中でファシリティを使用するわけだが，ファシリティは同時にひとつのトランザクションしか使用できないから，待たなければ使用できないこともある。そのため待ち行列キューができる」ということになる。どうです，わかりました？

それでは以上を頭にしっかり刻みこんだうえで表１を見てみよう。

まずシミュレーション時間。ほぼ3000秒といったところだ。これはこのシミュレーションが1000個のトランザクション（1000台のクルマ）を処理するまで，ということで実行されていることからも理解できるだろう。平均３秒に１台クルマがやってくるのだから，３秒×1000台は3000秒ということだ。

さて次はファシリティに関する結果である。最初はファシリティの番号。１，２，３はそのまま図の車線番号に対応している。ということはファシリティ１，２は右ハンドル専用の発券機，ファシリティ３は左ハンドル専用の発券機ということになる。２項目めのアベレージユーティリゼーションというのは「平均使用効率」という意味である。すなわちシミュレーション実行中，（時間比で）どれぐらい使用されていたかということを表している。

これを見るとシミュレーション実行中（約3000秒），ファシリティ１，２は99.5％も使用されていたことになる。逆にいえば，発券機が空いていたのは0.5％程度でしかない。

それに比べてファシリティ３は30％すらも利用されていない。ファシリティ１，２が混雑しているのをよそ目に，ファシリティ３はガラガラだったのである。

３項目めはシミュレーション終了までに何台のクルマがそれぞれのファシリティを使用したか，を表している。１，２は当然ほぼ同じで，合計で全体の９割ほど（＝右ハンドルの割合），３は左ハンドル専用であるから当然左ハンドルの割合１割にほぼ等しくなる。

ここで注意してほしいのだが，ファシリティ３の割合は設定の「左ハンドルは10％」には正確には合致しない。これはシミュレーションに乱数を用いているからである。考えてみればわかることだが，左ハンドルが10％だからといって，まず右ハンドルが900台きて，その次に左ハンドルが100台くるわけではない。お互いにごちゃまぜになっていて，全体として割合を見れば10％である，というふうになる。そこでコンピュータ上では乱数を用いて右/左ハンドルをうまく配分しているわけだが，乱数を用いている以上多少のゆらぎが出てきてしまうのである。

これは左ハンドルの割合だけの問題ではなく，結果すべてに共通することである。だから読者がリスト１を入力して実行しても表１とまったく同じ答えは出てこないだろう。傾向は同じでも多少の差は当然なのである（ましてこのような渋滞シミュレーションでは，渋滞が渋滞を呼んで大きな差を生ずることもある）。「あれ，ちゃんと入力したのに答えがあわない！」などとあわてないように。

さて，本題に戻るとしよう。次の項目はトランザクション１個（クルマ１台）を処理するのに要した時間の平均である。１，２，３ともほぼ７秒であるが，これは発券機の処理時間はどれも７±２秒である，という前提に合致している（正確に７秒にはならない理由は上で述べたとおり）。

さらに，最後に「in use」と表示されてい

表１　リスト１（改良前）の実行結果

```
Result

Simulation time (t) : 3092 s

 -- Facility --
   name        average        number      average
   or no.   utilization      entries    time / Xact
    1          0.9951          445        6.930              in use
    2          0.9942          440        7.002              in use
    3          0.2717          117        7.179


 --  Queue  --
   no.      max       average     total     zero    percent    average     current
         contents    contents    entries   entries   zero     time /Xact    contents
    1       17         7.283       460        5      0.011     48.952         15
    2       18         7.320       456        7      0.015     49.636         16
    3        2         0.037       117       82      0.701      0.966          0
```

るのは，そのファシリティが現在使用中であることを示している。

それでは次はキュー（待ち行列）の結果を見ていくことにしよう。まずは待ち行列番号であるが，これはそのままファシリティの番号に一致している。2項目めは最大どれぐらいの長さの待ち行列（現在使用中のトランザクションを除く，純粋に待っているトランザクションの数）ができたか，を示している。1，2はなんと最大で20台近くも並んだことになる。それに引きかえ3は最大でも2台しか待っていない。

3項目めは平均してどれぐらいの長さの待ち行列ができていたかである。前項は最大を表しているが，こちらは平均である。これによると1，2には大体いつも7，8台並んでいたことがわかる。一方3はといえば，ほとんど待ちなしだったことがわかる。

その次は待ち行列に入ったトランザクションの数である。前提から当然，約9対9対2の比率になっている。ファシリティのときの値より少し大きいのは，待っているうちにシミュレーションが終わってしまったクルマも含まれているから。

さて，次の2項目は待ち行列に入ったがファシリティが使用されていないので待ちなしで利用できたトランザクションの数とその割合である。容易に想像がつくように1，2ではほぼすべてのクルマが待たなければならないのに対して，3は70%は待ちなしという有様である。

その次はもっとも重要な結果，すなわち平均待ち時間である。1，2は平均で50秒近くも待たなければならないのに，3は1秒も待たない。なんたる不公平！

そして，ようやく，といった感じの最後の項目だが，これはシミュレーションが終了した時点でどれだけの待ち行列ができているか，を表している。かわいそうに1，2ではいまも15台ものクルマが待ち続けているのだ。

and...

さて，こうしてひととおりシミュレーションの結果を見てきた。どう感じただろうか。なになに左ハンドルだけが優遇されすぎている，って。ごもっとも，1割しかいない左ハンドルに1レーンまるまる提供しているのだから，さもありなん。

確かに結果から明らかなように，右/左ハンドルで平均待ち時間の格差が大きすぎる。これでは右ハンドルのドライバーが黙っていないだろう。

Ok，確かに君のいうとおり，ではどうするね。結果から問題点を指摘して終わりかい？

それもまたひとつの道だが，あまり賢明とはいえないだろう。ここで初心にたちかえって，なんのためにシミュレーションをするのかを考えてみよう。シミュレーションは目的があって行うもの。なにかのシステムのこんなところを改良したい。でも現実に調査して改善を図るのは効率が悪い。そんなときにこそシミュレーションで効率よく改善を図ろうとするのだ。

それではどうするか，もうおわかりだろうと思う。改善策を練り，シミュレーションで確かめてみよう。

上のシステムの欠陥は左ハンドルを優遇しすぎることにあった。これは第3レーンを左ハンドル専用にしてしまったからだ。それじゃあ第3レーンを左/右ハンドル兼用にしてみたらどうだろうか。第3レーンの右側にも発券機を設置すれば左右両側にあることになり，左ハンドルでも右ハンドルでも利用できる。

さて，図4が改良後の料金所である。プログラムはリスト1にリスト2をつぎはぎをするかたちで利用しよう。入力する際にはまずリスト1を入力し，改良に対応するためにリスト1のうち10070行から10140行までをリスト2のように書き換え，さらにリスト2の30000行以降をリスト1の最後に加えてほしい。はい，できあがり。

こうして得られたプログラムの実行結果

リスト1

```
1000 'Simulation model 2  ver.1.01
1010 '
1020 '        for X1 BASIC
1030 '
1040 '           1990.10  (c) Cammon
2000 '
2010 ' initialize : set constant
2020 WIDTH 80: CLS
2030 DEFINT a-z
2040 t!=0: nxact!=0
2050 pgen1=3: pgen2=2
2060 ptran=1
2070 p2(1)=7: p2(2)=7: p3(1)=2: p3(2)=2
2080 p2(3)=7: p3(3)=2
2090 oxact=1000
3000 '
3010 ' main
3020 REPEAT
3030  LOCATE 0,0: PRINT USING "  t  : #####";t!
3040  IF t!=nxact! THEN GOSUB 10020
3050  FOR i=1 TO 3
3060   GOSUB 20020: IF dxactt=oxact THEN i=3
3070  NEXT
3080  LOCATE 0,1: PRINT USING "Xact : #####";dxactt
3090  t!=t!+1
3100 UNTIL oxact=dxactt
3110 t!=t!-1
3120 GOSUB 25020
3130 END
10000 '
10010 ' generate Xact
10020 nxact! = t!+pgen1-pgen2+INT(RND(1)*(2*pgen2+1))
10030 xact=1
10040 IF INT(RND(1)*10) < ptran THEN p1=3: RETURN
10050 GOSUB 10070: RETURN
10060 '
10070 p1=INT(RND(1)*2)+1
10080 IF u(1)=0 AND u(2)=0 THEN RETURN
10090 IF u(1)=0 AND u(2) THEN p1=1: RETURN
10100 IF u(1) AND u(2)=0 THEN p1=2: RETURN
10110 IF q(1) = q(2) THEN RETURN
10120 IF q(1) > q(2) THEN p1=2 ELSE p1=1
10130 RETURN
10140 '
20000 '
20010 ' queue
20020 quwtt!(i)=quwtt!(i)+q(i)
20030 IF xact AND p1=i THEN GOSUB 20110
20040 IF u(i)=0 OR t!<qfreet!(i) THEN RETURN
20050 xacttt!(i)=xacttt!(i)+qfree(i)
20060 IF q(i) THEN q(i)=q(i)-1: GOSUB 20150 ELSE u(i)=0
20070 dxact(i)=dxact(i)+1: dxactt=dxactt+1: RETURN
20080 '
20090 'This program is dedicated to Saeko.Yes,it's you!
20100 '
20110 xact=0: nque(i)=nque(i)+1
20120 IF u(i) THEN q(i)=q(i)+1: GOSUB 20180: RETURN
20130 u(i)=1: nqze(i)=nqze(i)+1: GOSUB 20150: RETURN
20140 '
20150 qfree(i)=p2(i)-p3(i)+INT(RND(1)*(2*p3(i)+1))
20160 qfreet!(i)=qfree(i)+t!
20170 nfce(i)=nfce(i)+1: RETURN
20180 IF q(i)>maxq(i) THEN maxq(i)=q(i)
20190 RETURN
25000 '
25010 ' print result
25020 PRINT "Result": PRINT
25030 PRINT "Simulation time (t) :";t!;"s": PRINT
25040 PRINT " -- Facility --"
25050 PRINT "   name       average        number        average"
25060 PRINT "   or no.   utilization       entries   time / Xact"

25070 FOR i=1 TO 3
25080  PRINT USING "    #";i;
25090  PRINT USING "        ##.####";xacttt!(i)/t!;
25100  PRINT USING "          #####";nfce(i);
25110  PRINT USING "       ###.###";xacttt!(i)/dxact(i);
25120  IF u(i) THEN PRINT "            in use";
25130  PRINT
25140 NEXT
25150 PRINT: PRINT: PRINT " --  Queue  --"
25160 PRINT "   no.     max     average     total    zero   perc
ent   average    current"
25170 PRINT "           contents   contents  entries  entries  zer
o    time /Xact   contents"
25180 FOR i=1 TO 3
25190  PRINT USING "    #";i;
25200  PRINT USING "     ###";maxq(i);
25210  PRINT USING "      ###.###";quwtt!(i)/t!;
25220  PRINT USING "    #####";nque(i);nqze(i);
25240  PRINT USING "   ##.###";nqze(i)/nque(i);
25250  PRINT USING "   ###.###";quwtt!(i)/nque(i);
25260  PRINT USING "        ###";q(i)
25270 NEXT
25280 RETURN
```

が表2である。システム3と4で実行条件は基本的に同じである。違いは4では第3レーンが左右兼用になっているため，第3レーンがすいていれば右ハンドルのクルマもどんどん利用する，という点にある。

それでは実行結果を見てみよう。ひと目見ればわかると思うが，状況は劇的なまでに改善されている。平均待ち時間はどのレーンでも約2秒。改良前は50秒近かったのと比べると天と地ほどの差がある。これだけわずかな改良でこれほど効果があるとは正直いって筆者も予想していなかった。

ただ，今回は逆に第3レーンのほうが使用効率も高く，待ち時間も（わずかだが）長くなっている。これは第3レーンは右ハンドルに加え，1割いる左ハンドルも処理しなければならないからである。すなわち今度は逆に，左ハンドルのほうが待ち時間が長くなってしまったのである。

しかし左ハンドルの待ち時間が長くなったといっても微々たるものだし，多数派の右ハンドルの状況は大幅に改善されている。どちらのモデルのほうがよいかは明白だろう。当然後者だし，実際の料金所も後者のようになっている（はず）だ。

＊　　＊　　＊

以上，シミュレーションの実際例を見てきたわけだが，今回はどのようにシミュレーションを実現するか，というよりも，シミュレーションがどのように有効なのかを中心に考えてきた。シミュレーションをうまく使ってやれば，システムの改善の大きな力になることがわかってもらえたと思う。

ただ，こんな漠然としたモデルで，本当に現実を反映しているのだろうか，と考えている人もいるだろう。確かに今回のモデルは多分に漠然としていて，条件もあまり厳密なものとはいえない。でもこんなモデルでも，ちょっとの改良で劇的に状況が改善されることぐらいはわかる。まあ，これぐらいのモデルなら傾向がつかめれば十分だろう。

もし，もっと厳密な結果がほしいのであれば，条件をさらに厳密にしてやればよいだけのこと。その際でも原則はまったく同じである。

NEXT

今回は導入部ということで，シミュレーションの概略を見てきた。次回はどのようにシミュレーションを実現させるのか，から始めて，もっと複雑な例にも挑戦してみたいと思う。では，また。

図4　料金所の様子（改良後）

表2　リスト2（改良後）の実行結果

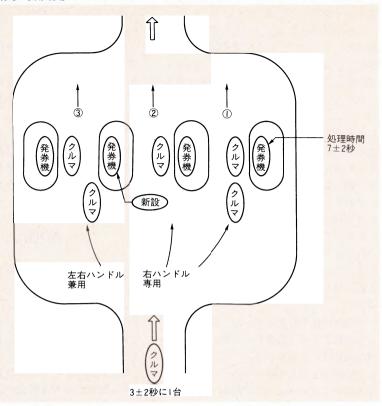
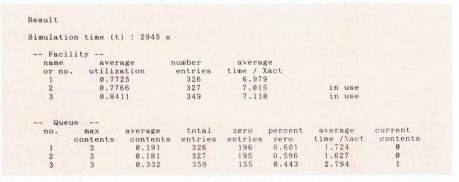
Result

Simulation time (t) : 2945 s

-- Facility --

name or no.	average utilization	number entries	average time / Xact	
1	0.7725	326	6.979	
2	0.7766	327	7.015	in use
3	0.8411	349	7.118	in use

-- Queue --

no.	max contents	average contents	total entries	zero entries	percent zero	average time /Xact	current contents
1	3	0.191	326	196	0.601	1.724	0
2	3	0.181	327	195	0.596	1.627	0
3	3	0.332	350	155	0.443	2.794	1

リスト2

```
10070 m(1)=u(1): m(2)=u(2): m(3)=u(3): GOSUB 30020
10080 IF flg THEN p1=min: RETURN
10090 IF min THEN ON min GOSUB 35010,35020,35030: RETURN
10100 IF u(1)=0 THEN GOSUB 35040: RETURN
10110 m(1)=q(1): m(2)=q(2): m(3)=q(3): GOSUB 30020
10120 IF flg THEN p1=min: RETURN
10130 IF min THEN ON min GOSUB 35010,35020,35030: RETURN
10140 GOSUB 35040: RETURN


30000 '
30010 ' minimum routine
30020 IF m(1)<m(2) AND m(1)<m(3) THEN min=1: flg=1: RETURN
30030 IF m(1)<m(2) AND m(1)>m(3) THEN min=3: flg=1: RETURN
30040 IF m(1)<m(2) AND m(1)=m(3) THEN min=2: flg=0: RETURN
30050 IF m(2)<m(3) AND m(2)<m(1) THEN min=2: flg=1: RETURN
30060 IF m(2)<m(3) AND m(2)>m(1) THEN min=1: flg=1: RETURN
30070 IF m(2)<m(3) AND m(2)=m(1) THEN min=3: flg=0: RETURN
30080 IF m(3)<m(1) AND m(3)<m(2) THEN min=3: flg=1: RETURN
30090 IF m(3)<m(1) AND m(3)>m(2) THEN min=2: flg=1: RETURN
30100 IF m(3)<m(1) AND m(3)=m(2) THEN min=1: flg=0: RETURN
30110 min=0: flg=0: RETURN
35000 '
35010 p1=INT(RND(1)*2)+2: RETURN          'generate 2 or 3
35020 p1=(INT(RND(1)*2))*2+1: RETURN      'generate 1 or 3
35030 p1=INT(RND(1)*2)+1: RETURN          'generate 1 or 2
35040 p1=INT(RND(1)*3)+1: RETURN          'generate 1 or 2 or 3
```

★(で)のショートプロぱーてぃ その15

テクニックは偉大なのだ!

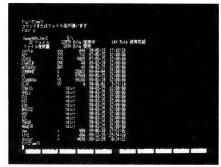
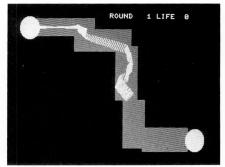
Komura Satoshi 古村　聡

とりあえず「3Dダンジョンを描く」ということにテーマも決まり，今月からいよいよぱーてぃハンズの第2部が始まります。今後ともよろしく。ショートプログラムはX68000の画面クリアツールとX1のアクションゲームです。

illustration : T.Takahashi

いやあ，先月はお休みになってしまったわけですが，別に連載が打ち切りになってしまったのではないのでご安心を。まあ，ほかのページにはちょこちょこと登場していたのでファンの皆さんもさびしくはなかったでしょう（ファンなんかいないって？グッスン)。

さて，季節はすっかり秋深し，という感じですよねー。みんなどうやって時間つぶしているんでしょうか。私の場合はやっぱりプログラミングとかパソコンに向かいっぱなしということなんですけど……。少しわびしい。

しかし，それだけというわけじゃなーい。私には原稿書きという天から与えられた使命がある（そんなもの与えた覚えはないとかいわれそうだけど)。特に，この2，3カ月というものはハンズのネタに苦しんでいたんです。まぁ，それもハンズを読めばわかるようにある読者の方によってなんとかなりました。つくづく，他人任せな（で）だなあと自分のことながら思ったりするので

とろける.X

KURUPER

リスト1　とろける.X

```
 1: *        テキスト画面が溶ける Ver 1.01
 2: *                             M E L T
 3: *                                1990/08/15
 4: *                                By T.A.K
 5:
 6: .include          doscall.mac
 7: .include          iocscall.mac
 8:
 9:                   .text
10:                   .even
11: inimlt:
12:                   lea.l   w_fnt,a1
13: start:
14:                   dc.w    $fe0e             *乱数発生 d0
15:          *＄ＦＥＯＥ　＿＿ＲＡＮＤ
16:          *返り値　乱数（ロングワード符号付き）
17:          *このコールはfloat2 or 3が必要です。
18:
19:                   move.l  d0,d3             *乱数をＤ３へ
20:                   and.l   #$3ff,d3          *Ｄ３を0～767に
21:                   and.w   #$100,d0
22:                   beq     go
23:                   and.w   #$1ff,d3
24: go:
25:                   move.l  #1,d1             *テキスト設定
26:                   IOCS    _TCOLOR
27:                   move.l  #$e0ff80,vraddr *set vram addr
28:                   move.l  d3,d1             *Ｘ座標設定
29:
30:                   bsr     zurasi1           *プレーン１
31:                   bsr     zurasi2
32:                   bsr     zurasi1
33:
34:                   moveq.l #2,d1
35:                   IOCS    _TCOLOR
36:                   move.l  #$e2ff80,vraddr *
37:                   move.l  d3,d1
38:                                             *プレーン２
39:                   bsr     zurasi1
40:                   bsr     zurasi2
41:                   bsr     zurasi1
42:
43:                   IOCS    _B_KEYSNS         *押されたか?
44:                   tst.w   d0
45:                   beq     start
46:                   move.w  #16,d1
47:                   move.w  #2,-(sp)
48:                   move.w  #10,-(sp)
49:                   DOS     _CONCTRL          *画面をきれいに
50:                   move.w  #0,-(sp)
51:                   move.w  #14,-(sp)
52:                   DOS     _CONCTRL
53:                   DOS     _EXIT             *終わり
54:
55: *ブロック転送
56: *in (a2)
57: *break (a2)+
58: zurasi2:          clr.w   d2
59:                   IOCS    _TEXTGET
60:                   move.w  #2,d2
61:                   bra     puttx
62:
63: zurasi1:
64:                   clr.w   d2
65:                   IOCS    _TEXTGET          *vram内容取る
66:                   move.w  #1,d2             *Ｙ座標設定
67: puttx:            IOCS    _TEXTPUT          *テキストプット
68:                   addq.w  #1,d1             *x++;
69:                   rts
70:
71: vraddr:           .dc.l   $e00000
72: w_fnt:            .dc.w   1,511
73:                   .ds.w   64
74:
75:                   .end
```

あります。

さて，たわいもない話はこれぐらいにして（？）本題のほうをぼちぼちいきますか。

画面がドロりん

ではでは，今月の１本目。まずはX68000用の画面消去ユーティリティ「とろける．X」です。そう，字が溶けるんです。

とろける．X　for　X68000

（要アセンブラ，リンカ）

広島県　一岡孝浩

このプログラムはアセンブラで書かれていますので，実行するためにはアセンブラ，リンカ，それとDOSコール，IOCSコール用のマクロが必要になります。アセンブラ，リンカは福袋や，C compiler PRO-68K Ver.1，Ver.2などに入っています。ただしマクロは（iocscall.mac，doscall.mac）は福袋にはついてきませんので，C compiler PRO-68KあるいはOh!X 6月号の付録のディスクに入っているものを使用してください。

プログラムを実行するには，まず，

A>ed とろける.S

としてエディタを立ち上げます。そしてリスト１を打ち込んでいってください。あ，行番号は入れちゃだめですよ。打ち終わったら，

[ESC]・E（ESCキーを押してからEを押す）

としてセーブしてから，エディタを終了してください。

続いて，アセンブル&リンクをします。

A>as とろける

A>lk とろける

アセンブル，およびリンクが終わりました。エラーやウォーニングは出ていませんね？　出ていたら打ち間違いがあると思われますから，間違いを見つけて直してください。あ，あと，.macファイルがないとエラーになりますから，ちゃんとプログラムと同じディレクトリに置いてください。

これですべてよし，と。あ，そうだ実行させる前に効果がわかりやすいように画面に文字を出しておきましょう。まあ，

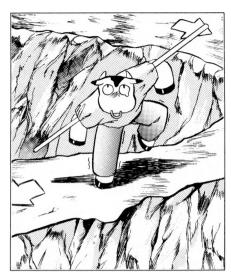
A>DIR

とかしてディレクトリを表示しておけばいいかな。はい，それでは，

A>とろける

で実行させてみましょう。……おお，溶けてゆく。適当なところでスペースキーを押すと画面がクリアされます。溶けるっていえば，PDSにもMELT.X（tarazo lab氏作）

リスト２　とろける．Xの変更点

```
 56: *ブロック転送
 57: *in (a2)
 58: zurasi2:
 59:                 clr.l   a1
 60:                 IOCS    _B_SUPER
 61:                 move.l  d0,a6
 62:                 bsr     xyaddr
 63:                 move.w  #102,d2
 64:                 move.l  #-256,a4
 65: loop51:         bsr     loop512
 66:                 subq.w  #1,d2
 67:                 bne     loop51
 68:                 addq.w  #1,d1               *x++;
 69:                 move.l  a6,a1
 70:                 IOCS    _B_SUPER
 71:                 rts
 72:
 73: zurasi1:
 74:                 clr.l   a1
 75:                 IOCS    _B_SUPER
 76:                 move.l  d0,a6
 77:                 bsr     xyaddr
 78:                 move.w  #102,d2
 79:                 move.l  #-128,a4
 80:
 81: loop5:          bsr     loop512
 82:                 subq.w  #1,d2
 83:                 bne     loop5
 84:                 addq.w  #1,d1               *x++;
 85:                 move.l  a6,a1
 86:                 IOCS    _B_SUPER
 87:                 rts
 88:
 89: loop512:        move.b  0(a3,a4),d4
 90:                 and.b   d0,d4
 91:                 beq     prese2
 92: pset2:          or.b    d0,(a3)
 93:                 bra     decad2
 94: prese2:         eor.b   #$ff,d0
 95:                 and.b   d0,(a3)
 96:                 eor.b   #$ff,d0
 97: decad2:         lea     -$80(a3),a3
 98:
 99:                 move.b  0(a3,a4),d4
100:                 and.b   d0,d4
101:                 beq     prese21
102:                 or.b    d0,(a3)
103:                 bra     decad21
104: prese21:        eor.b   #$ff,d0
105:                 and.b   d0,(a3)
106:                 eor.b   #$ff,d0
107: decad21:        lea     -$80(a3),a3
108:
109:                 move.b  0(a3,a4),d4
110:                 and.b   d0,d4
111:                 beq     prese22
112:                 or.b    d0,(a3)
113:                 bra     decad22
114: prese22:        eor.b   #$ff,d0
115:                 and.b   d0,(a3)
116:                 eor.b   #$ff,d0
117: decad22:        lea     -$80(a3),a3
118:
119:                 move.b  0(a3,a4),d4
120:                 and.b   d0,d4
121:                 beq     prese23
122:                 or.b    d0,(a3)
123:                 bra     decad23
124: prese23:        eor.b   #$ff,d0
125:                 and.b   d0,(a3)
126:                 eor.b   #$ff,d0
127: decad23:        lea     -$80(a3),a3
128:
129:                 move.b  0(a3,a4),d4
130:                 and.b   d0,d4
131:                 beq     prese24
132:                 or.b    d0,(a3)
133:                 bra     decad24
134: prese24:        eor.b   #$ff,d0
135:                 and.b   d0,(a3)
136:                 eor.b   #$ff,d0
137: decad24:        lea     -$80(a3),a3
138:
139:                 rts
140:
141: *VRAM　アドレスの計算
142: *inp d1=X座標
143: *out (a3)=VRAM　アドレス
144: *     d0=マスクパターン
145: xyaddr:
146:                 move.l  d1,d0
147:                 move.l  d0,d5
148:                 lsr.l   #3,d0
149:                 add.l   vraddr,d0
150:                 move.l  d0,a3
151:                 and.l   #%111,d5
152:                 lea.l   mskpat,a0
153:                 move.b  0(a0,d5.l),d0
154:                 rts
155:
156: mskpat:         .dc.b   %10000000
157:                 .dc.b   %01000000
158:                 .dc.b   %00100000
159:                 .dc.b   %00010000
160:                 .dc.b   %00001000
161:                 .dc.b   %00000100
162:                 .dc.b   %00000010
163:                 .dc.b   %00000001
164:
165: vraddr:         .dc.l   $e00000
166: w_fnt:          .dc.w   1,511
167:                 .ds.w   64
168:
169:                 .end
170:
```

とかがありますよね。こっちのほうが不気味だけど。

アセンブラはいいぞ！

作者の一岡さんはアセンブラでプログラムを作ったのはこれが初めてなんだそうです。拍手拍手。投稿原稿によると“本当は直接テキストVRAMをいじりたかったのですが……”ということです。なるほど，これはIOCSを使っているのか。それじゃあってことで私，早速やってみました。リスト2がI/O直接叩くバージョンへの変更点です。リスト1の55行から69行までをリスト2と入れ替えてみてください。

えっと，なにをしているかというとオリジナルではIOCSにお願いして，ランダムに選んだX座標から，

X＋0　縦に1ドットずり下げる

X＋1　同じく2ドット下げる

X＋2　もう1回1ドット下げる

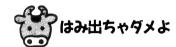
ということを繰り返して垂れているように見せているわけです。こいつをスーパーバイザモード（これが直接I/Oをいじるモードね）で自分でしこしこと1ドットずつずり下げるようにしたんです。結構スピードも速くなったでしょ（期待してたよりは遅かったんだけど。まだまだ精進が足りないかな？）。

いやあ，やっぱりこれだよね，これ。マシン語の醍醐味はI/Oに始まりI/Oに終わるのです。自分のための自分のパソコン，やっぱり骨の髄までしゃぶってしまいたい。機能の隅から隅までズズ，ズイーっと使い切ろうとするにはやっぱり小回りの効くマシン語がベストなんですよ。

やっぱりマシン語は速い！　それにちょっと実行の順序を変えただけでも露骨にスピードが変わってきたりして一番面白い言語なんです，実は（と私は思う）。だから，みんな“スピードアップのテクニック”なんてのを極めてやろうとあがいたりしているわけだ，これが。

そういう世界であるからして，当然のように金言格言，先人の知恵というのも多数存在していて，いまでも「ループ展開は勝利！」「案ずるよりイミディエイトは易し」「SUPERで万全」……。ばかもの！　偏ったことを教えるんじゃない！　どうせなら「載ってるものはROMでも使え」とか……。いや，それも偏ってるんじゃない。うーん，ひとりでなにいってんだろう。それにマシン語がわからんとちっとも面白くないだろうな。

ちなみに，リスト2も結構偏ってます。さすがに512回ループ展開&場合分け4回なんていう暴挙には出なかったけどね。

はみ出ちゃダメよ

なにがなんだかわからなくなったところで（初心者を陥れる入門講座なんて，そうそうないぞ），今月の2本目いきます。X1用のアクション（？）ゲーム，KURUPER（クルッパー）です。

KURUPER for X1シリーズ

(CZ-8FB01)

神奈川県　森貴之

新旧どちらのBASICでもOKです。画面上に■（ボックスフィルといえばいいのだろうか？）が並んで道ができています。スタート地点をくるくる回っているのが主人公のクルッパー君です（ただのラインにしか見えない？　いいの，ただのラインだもん）。くるくる回転してるからクルッパーというわけ。このクルッパー君を道からはみ

リスト3 KURUPER

```
10 CLS4:INIT:WIDTH 40:CLICK OFF:LI=2:LEV=.02
20 LOCATE12,12:PRINT"K U R U P E R !!":A$=INKEY$(0)
30 IF A$="" THEN 20 ELSE M=VAL(A$):IF M<1 OR M>5 THEN M=1
40 CLS4:O=100:ZX=0:ZY=0:GOSUB310
50 ON M RESTORE 330,350,390,430,470
60 READ A:FORI=1TOA  STEP 4:READ A1$,A2$,A3$,A4$
70 A1=VAL("&H"+A1$):A2=VAL("&H"+A2$)
80 A3=VAL("&H"+A3$):A4=VAL("&H"+A4$)
90 LINE(A1+32,A2)-(A3+32,A4),PSET,1,BF:NEXT I
100 CIRCLE(20,20),16,6:PAINT(20,20),6
110 CIRCLE(300,180),16,6:PAINT(300,180),6:X=20:Y=20
120 O=O+LEV:S=STICK(0)+STICK(1)
130 IF STRIG(0)+STRIG(1) THEN ZX=0:ZY=0
140 ZX=ZX+(S=4)-(S=6):ZY=ZY+(S=8)-(S=2)
150 C=6:GOSUB 230:X=X+ZX:Y=Y+ZY
160 X1=COS(O):Y1=SIN(O)
170  IF POINT(X-X1*8,Y-Y1*8)=0 THEN 240
180  IF POINT(X+X1*8,Y+Y1*8)=0 THEN 240
190  IF POINT(X,Y)=0 THEN GOSUB 240
200 C=2:GOSUB 230
210 IF X>295 AND Y>175 THEN 290
220 GOTO 120
230 LINE(X+X1*8,Y+Y1*8)-(X-X1*8,Y-Y1*8),PSET,C:RETURN
240 C=2:GOSUB 230:
250 IF LI=0 THEN 270
260 LI=LI-1:LOCATE14,12:PRINT"MISS !!":PLAY"O4C3":PAUSE5:
GOTO 40
270 LOCATE15,12:PRINT"GAME OVER":PLAY"04C3DECDEDDDCC"
280 IF STRIG(0)+STRIG(1) THEN RUN ELSE 280
290 LOCATE17,12:PRINT"CLEAR":LI=LI+1:M=M+1:IF M>5 THEN M=
1 LEV=LEV+.02
300 PLAY"O4E2DCEDDDC":PAUSE10:GOTO 40
310 COLOR7:LOCATE18,0:PRINT" ROUND  ";M;"LIFE "LI:RETURN
320 '— ROUND   1 ————————————————————
330 DATA 20, 00,07,5F,20, 3E,17,9D,36, 83,2F,A8,7B, 90,7B
,B5,BF, B1,A2,FF,C4
340 '— ROUND   2 ————————————————————
350 DATA 60, 00,0A,13,1C, 14,0C,AE,1A, AF,0A,C8,2B, C8,18
,E1,40, 69,34,D5,45
360 DATA      1A,40,86,51, 04,49,24,74, 1C,5F,3C,93, 35,7A
,7E,9E, 6D,6F,B6,83
370 DATA      98,7E,E1,92, BC,91,E1,A5, A3,A3,C8,B7, C4,AE
,E9,C2, EA,AA,FF,BE
380 '— ROUND   3 ————————————————————
390 DATA 44, 00,0B,26,1E, 12,1E,41,31, 38,0A,67,1D, 55,1E
,84,31, 74,32,A3,45
400 DATA      95,46,C4,59, 80,5A,AF,6D, 60,6E,8F,81, 7D,82
,AC,95, 9B,96,CA,A9
410 DATA      B4,A9,FF,BE
420 '— ROUND   4 ————————————————————
430 DATA 44, 00,07,1A,22, E6,07,FE,22, E6,A8,FE,C3, 00,A8
,18,C3, 00,57,18,72
440 DATA      E7,57,FE,72, 1A,10,E6,19, 19,60,E6,69, 19,B2
,E6,BB, EB,1A,FB,5E
450 DATA      04,6B,14,AF
460 '— ROUND   5 ————————————————————
470 DATA 40, 04,10,54,18, 47,15,97,25, 89,22,DB,2A, B2,B0
,FB,B8, 73,A3,BC,B3
480 DATA      33,9E,7C,A6, 21,59,33,A6, DC,22,EE,6F, 21,59
,88,63, 75,64,DC,6F
```

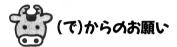
出ないように2，4，6，8のキーでゴールまで運んであげてちょうだいな。クルッパー君は細長いけど，くるくる回転してるからタイミングがすべて。広いところで待っていて，角度がよくなったら一気に道を渡っちゃえ！

そうそう，ひとついい忘れてた。クルッパー君には慣性が働いているので，あんまり勢いよく渡ろうとすると止まれなくなるので注意して……。ああ，遅かったか。面クリアで1UP，また，タイトル画面で1～5のキーを押すと面セレクトができます（つまり全5面あるのだ）。

慣性か，そういや最近は慣性を使ったものが多いなあ。さすがにパズルとかは別みたいだけど，慣性のかかるパズルってあったかなあ。でも，いったいどんなものになるんだろ。

ところでこのゲームで遊んでいたら，「これショート？　見た目にきれいだね」「写真うつりよさそうな画面ですね」とスタッフの間でもなかなかの評判でした。うーん，テクニシャン。ラインの魔術師と呼んであげよう。ラインがくるくる回転しながら動いて，しかも残像みたいに軌跡が残っていくから，ヒラヒラ舞っているように見えるんだよね。本当にいいので，ぜひとも打ち込んでみてください。

画面センスがいいっていうのも大切な要素ですよね。短いプログラムで画面をきれいに見せるのは難しいですけどね。センスを磨くのもテクニックのひとつかな。

(で)からのお願い

ううう っ。実は「とろける．X」の一岡さん，プログラムをディスクじゃなくてプリントアウトで送ってきてくれたのです。薄い封筒に（で）様なんて書いてあるからファンレターかと思ったじゃないですか（来るわけないか）。

あのね……，別に打ち込むのが面倒臭いとはいわないけど（そんなことといったら読者の皆さんに申し訳ないもんね），私が打ち込むんだよ。タイプミスしてエンバグ（デバッグの逆で人のプログラムにバグを付け加えてしまうこと）しちゃうかもしれないんだよ。それが全国何十万，何百万部（かなりオーバーだな）も売れていくんだよ。恐ろしいでしょう？

だから，皆さんディスクで送ってきてください。ディスクの容量が余ってもったいないっていうんだったら，いっぱいPCMデータ入れてもかまわないし（あんまり意味がない？），制作日誌などを入れてもいいんですから。

しかし，本当にエンバグしてないかな。なかなか不安な今月の（で）なのでありました。どっかに打ち間違いがあったらこっそり教えてね。まあ，ほとんど（絶対？）大丈夫だと思いますがね。そんなこんなでまた来月。

精進せえよ。

(で)のぱーてぃハンズ第2部———(その1)

はい，皆さんお待たせしました（だれも待ってないって!?　まあ，そういわずにお茶でもどうぞ)。ぱーてぃハンズ第2部のスタートです。

ううっ，やっと来ましたリクエスト。うれしいじゃありませんか。愛知県の白井達広さん，ありがとね。

では，さっそく読ませていただきます。なになに，

「“1周年特別企画－どんちゃん騒ぎの部屋”にてリクエストよろしくと書かれているのに気づいたので考えました。

1)　ウィ○ードリ○ィのようなラインで構成された（壁でもいいけど）の3D迷路のRPGを作る（モンスターは出なくてもいい）

2)　コマンド選択式のテキストアドベンチャーを作る

3)　マップを配列変数に詰め込んだスクロールするRPGのようなものを作る」

ふむ，どれも面白そうですね。1番はあれかな。要するにダンジョンとか3D迷路を描くプログラムを作って，あとはイロイロと付け足せばいいのかな。これがいいか。せっかく1番目に書いてくれたことだし。

てなわけで第2部は「ダンジョンを描くのだ」というセンでいきましょう。ネタが決まって，めでたし，めでたし。

方針なのである

まずはなにをどういうふうに作っていくか決めずばなるまい。作るのは3Dのダンジョン。これを配列上のデータにしたがって，画面上に描いていくとかすればいいのかな。で，マップデータなんだけどこいつはいろいろな種類を作って，ラインの色を変えたりメッセージが出せるようにしておこう。そうすれば将来的にはいろんなイベントも入れられるし，階段を作ったりもできるでしょ。今回は1階分のデータだけでほかの階や実際のイベントは各自で作っていただきましょう。あ，モンスターは出なくていいってことなんでモンスターもなしね（こらこら)。

よしよし，だいたいの方針は立った。ところでなにから作ろうか？　こういう場合は一番簡単そうで，しかも目に見えるところから手をつけるのが得策だったりするのです。なんでかっていうと，難しくって目に見えないところからやっちゃうと，バグが出た場合に修正が鬼ムズになっちゃうからなのです。

だって，実行結果が目に見えないんですよ。プログラムのどこが間違ってるのかプログラムリストとにらめっこ（まあ，どんなプログラムのデバックも最終的にはこれをやるんだけどね）しなくちゃバグが絶対わからない。おまけに打ち間違いくらいのバグならいいけどハナからアルゴリズムを間違えてたなんてシャレにもならない事態になったらそれこそ悲惨だものね（まあ，簡単なところでもアルゴリズムが間違っているとシャレにならんかもしれないけど)。

さて，3Dダンジョンを描くにあたって，ダンジョンを描くステップを1つひとつ順に書いていくと，

　　自分のいる位置を確認する

　　↓

　　自分のいる位置からどのくらいのところになにが見えるのかを調べる

　　↓

　　自分の位置から見えるところにあるものを描く

となるわけだな。こいつを，

1)　壁を描くルーチン

2)　自分から見える壁を選んで1)に壁を描かせるルーチン

3)　自分の位置から向いている方向を調べ，その方向にある一定の位置の壁を見えるものとみなして2)に壁を描かせるルーチン

という順番で作っていくと，実際に壁を描く様子を少しずつ見ながらデバッグができるのだな。

おお，これは目に見える部分から作っていく法則にしたがっているではないか。

こいつはボトムアップ法という名前でよく知られているプログラムの組み方だったりするので（ボトム，下位のルーチンから徐々に上のルーチンに上がる，アップしていくわけだな），実はえらくもなんともなかったりするのだが……。

おお，重要なことを忘れていた。今回使用するのはＸ68000，言語はX-BASICなのであった。やっぱりなんだかんだいってもBASICが一番デバッグしやすいということと，初心者だろうがなんだろうがＸ68000ユーザーが全員持っているからという実にリーズナブルな選択なのである。一応，初心者向けということなので出来もスッポンと手抜きであることだし，中級者以上の方はアルゴリズムの説明だけ読んで自分できっちりとプログラムを組み，コンパイルするなりＣに移植するなりしていただきたい。とりあえずそういうこと。

さっそく説明開始

では，実際のプログラムに入ります（最初だからといってプログラムに入らないほど私は甘くないのだ。まるで学校の意地悪な先生だなやることが)。まずは目に見えるものということだから，ダンジョンのパターンを全部描き出してしまいます。

最初に考えなくてはならないのが「何歩先まで見えることにするか」ということ。つまりバリバリ全開に視界が開けてててもある程度先以上は見えなくしてしまうというわけなのね，これが。本当は無限の彼方に壁があったら地平線になるようにできればいいんだけどそこまでやってもあまり意味はない，ということである程度以上は見えないということにしてしまうわけ。で，ここでは正面が４段階，横の壁は３段階まで見えるということにします。これをマップの状況に応じて組み合わせて壁を描いていくわけね（うーん，１対１対応。グラフ理論だな)。

あ，あと絵を描く前にすることがあった。画面の初期設定ってやつです。うんと，これはなにかというと，これからグラフィックで絵を描くわけですから、“これから絵を描くぞー”という宣言をしてやらないかんわけです。

それにですね。画面モードというやつがありまして，なにを使いたいかによっていろいろ設定しなくちゃいかんのですよね。で、とりあえず、画面に絵を描くぞー、画面は512×512ドットで16色使えるモードにするぞー、ということを宣言するのにはSCREEN文というのを使います。SCREEN文に関してはBASICのマニュアルを参照のことね(SCREENに限らずわからない命令があったらマニュアルを見ようね)。

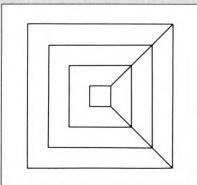
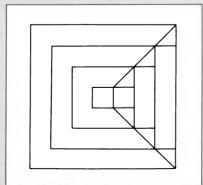
はい、おまたせ。それではみんなで壁のパターンのお絵描きをしてみましょう。まず、画面一杯にボックスするようなパターンを作ります。それから少しずつ小さくなるように描いていくんですね。小さくなるパターンをあと３つ、ボックスで描いてやります。そうそう、要するに遠近法ってやつです（あ、そうだ。本当の遠近法では遠くにいくほど壁の間隔は狭くなるんですが……。はっ、みんな等間隔になってる。ま、それでもちゃんと見えるからいいよね)。

次にサイドのパターンをさっき作った□にあわせて斜めの線をひとつずつ引いていきます。図１のような感じですね。右側全部描けました？　それじゃ、壁が途中で切れているパターンね。横に壁がないところは図２のように横線を引きます。できたかな。あとは左にも同じようにするだけです。座標に気をつけてね。

おっと、忘れるとこだった。実際にはメッセージウィンドウのために画面下と右側に少しスペースを空けている。だから、正確には“ダンジョンの絵のエリア一杯のボックス”なのだ。よろしい？

というわけで壁のパターンができた。正面４パターン、サイド３×２（左右）パターン、サイドの切れ目３×２パターン。全部で16個の壁を描いたわけだ。ふーっ、さすがに私もつかれた。リストも結構長いし。

では、今月はこのへんにしておきましょう。おなかもへったことだし。来月はこれを使って目に見える範囲の壁を描くルーチンを作ります。とりあえず、

drawbox1( )

とか、

drawsl2( )

とかやってみてどんな絵が描けるか試してみてください。

では，また来月このOh!Xで。ぱっくん(とあんぱんを食べて去る)。

図１

図２

リスト

```
 280 /*その他の初期化*/
 290 clrscr()
 380 end
1070 /*画面の初期化*/
1080 func clrscr()
1090 screen 1,1,1,1
1100 box(0,384,511,511,15)
1110 locate 0,24
1120 print"運命の迷宮へようこそ"
1130 endfunc
2550 func drawbox1(wcol)
2560 box(0,0,383,383,wcol) /*深さ1*/
2570 endfunc
2580 func drawbox2(wcol)
2590 box(48,48,335,335,wcol)/*   2 */
2600 endfunc
2610 func drawbox3(wcol)
2620 box(96,96,287,287,wcol)/*   3 */
2630 endfunc
2640 func drawbox4(wcol)
2650 box(144,144,240,240,wcol)/* 4 */
2660 endfunc
2670 func drawsl1(wcol)
2680 /*左深さ1*/
2690 line(0,0,47,48,wcol)
2700 line(47,48,47,335,wcol)
2710 line(48,335,0,383,wcol)
2720 /*line(0 ,383,0,0,wcol)*/
2730 endfunc
2740 func drawsl2(wcol)
2750 /*左     2*/
2760 line(48,48,95,96 ,wcol)
2770 line(95,96,95,287,wcol)
2780 line(95,287,48,335,wcol)
2790 line(48,335,48,48,wcol)
2800 endfunc
2810 func drawsl3(wcol)
2820 /*左     3*/
2830 line(96,96,143,144,wcol)
2840 line(143,144,143,240,wcol)
2850 line(143,240,96,287,wcol)
2860 line(96,287,96,96,wcol)
2870 endfunc
2880 func drawsr1(wcol)
2890 /*右     1*/
2900 line(383,383,336,335, wcol)
2910 line(336,48,383,0,wcol)
2920 line(383,0,383,383,wcol)
2930 line(336,336,336,48,wcol)
2940 endfunc
2950 func drawsr2(wcol)
2960 /*右     2*/
2970 line(335,335,288,287,wcol)
2980 line(288,287,288,96,wcol)
2990 line(288,96,335,48,wcol)
3000 line(335,48,335,335,wcol)
3010 endfunc
3020  func drawsr3(wcol)
3030 /*左     3*/
3040 line(287,287,241,240,wcol)
3050 line(241,240,241,144,wcol)
3060 line(241,144,287,96,wcol)
3070 line(287,96,287,287,wcol)
3080 endfunc
3090 func drawnl1(wcol)
3100 /*左      1*/
3110 box(0,48,47,335,wcol)
3120 endfunc
3130 func drawnl2(wcol)
3140 /*左      2 */
3150 box(48,96,95,287 ,wcol)
3160 endfunc
3170 func drawnl3(wcol)
3180 /*左      3 */
3190 box(96,144,143,240,wcol)
3200 endfunc
3210 func drawnr1(wcol)
3220 /*右      1*/
3230 box(383,335,336,48,wcol)
3240 endfunc
3250 func drawnr2(wcol)
3260 /*右      2 */
3270 box(335,287,288,96,wcol)
3280 endfunc
3290 func drawnr3(wcol)
3300 /*右      3 */
3310 box(287,240,241,144,wcol)
3320 endfunc
```

X68000用 ©KONAMI グラディウスIIIより Sand Storm

Kashiwagi Katutoshi
柏木 勝利

X1/turbo用 ©システムサコム/Yonao Keishi メタルサイトより Into The Shadow

Takahashi Tetushi
高橋 哲史

木の葉も黄色くなって，もう街はすっかり秋の色。吹いてくる風もちょっぴり冷たくなってきました，みなさんお元気ですか？
さて，今月のLIVE inは，そんな落ち着いた情景とはうらはらに，ノリのいいゲームミュージック2本立てで攻めてみました。どちらも力作，ぜひ打ち込んで聴いてみてくださいね。

やはりきたか，グラIII

X68000用にはグラディウスIIIからステージ1の“Sand Storm”をお送りしましょう。グラディウスといえば，X68000とは切っても切れない関係にあるシューティングゲームでしょう。その後継であるグラディウスIIIは，マニアのためにあったとまでいわれたゲームで，あまり普通の人向きではなかったようです。

さて，この作品は見てもらえばすぐにわかることとは思いますが，Yコマンドのてんこ盛りになっています。作品の完成度を高めるためにはしょうがないことかもしれませんが，やはり入力する人のことを考えるとあまりお勧めできません。西川善司さんのように，サブルーチンを作って展開するほうがスマートになると思いますよ。関数もきっちりと煮詰めれば，Yコマンドの行列と同等の効果は得られることでしょう。サンプリングだけでも変数に定義していたのが，せめてもの救いというものです。

まあ，それだけ凝っただけあって見事に採用となりました。プログラムを作る人も大変だったのでしょう。苦労話が同封の手紙につらつらと書き連ねてありました。プログラムは，よく見てみると使い回しがきく行があるようなので，そこいらへんを最初にチェックしてから入力すると，少しは楽になるかもしれません。

このリストを入力した人へのリクエストなのですが，ぜひOPMAではなく，OPMDで聴いてください。なぜなら，サンプリングされているドラム音が，すべてKORGのM1からサンプリングされているOPMAに対し，OPMDではM1のほかに一部で専用のドラムマシンの音を用いているからなのです。そのせいか，OPMDで聴いたほうが曲により迫力が加わるようです。

グラディウスIII

メタルサイト

ミッション・メタルサイト

先月号で，スタッフが作ればどんな機種でも毎月掲載できるんだぞ！　と書いたのは，皆さんを叱咤激励するつもりだったのです。ところが，その原稿を締め切りギリギリに持ち込んだ私を待っていたのは，お絵描きスタッフの高橋君から届いていた投稿だったのです……。

断っておきますが，頼んで作ってもらったのではありません。あくまでも投稿扱いをしています。いままで投稿をしたことがある人ならわかると思いますが，きっと彼のもとにもOh!X特製の記念品が届くことでしょう。十分な審査のうえ，今月送られてきた曲のなかではいちばんよくできていたので採用になりました。

曲は，X68000専用シューティングゲーム“メタルサイト”のステージ1の曲，Into The Shadowです。このゲームは全体的にレベルが高くて，グラフィックや操作性はもちろん，BGMもかなり秀逸なデキなのです。その移植とあれば十分に難しいと考えられます。まあ，音源の数を単純に比較してしまえば，X1のほうが有利ではありますが，やっぱりサンプリングの有無の影響は，結構大きいものでしょう。

ところがこの作品は，**初めてMMLをさわった**人が作ったとは思えないくらいによくできています。もともと，高橋君はお絵描き専門だったはず。ましてや「いままで音楽プログラムを作ったことはなかったんですよ」とは本人の弁。それでもここまでできるんですねぇ。きっとよいアドバイザーがいて，意見してくれたんだと思います。皆さんも，作品を作ったらすぐに投稿しないで，友達や兄弟など，無理やりにでも聴かせていろいろと評価してもらいましょう。意外と自分では気がつかなかった音のはずれなどが見つかったりしますよ。

皆さんも見よう見真似でもいいんです，チャレンジしてみてください。ひょっとすると来月号のLIVE inを飾るのは，あなたの作品かもしれませんよ。　　(S.K.)

リスト1　グラディウスIII

```
10 /*            save"GRADIUSIII.ST1
20 /*
30 /*              G R A D I U S I I I
40 /*
50 /*   S a n d   S t o r m  ( Stage 1 ・ 砂塵 )
60 /*
70 /*                 作曲・編曲 コナミ矩形波倶楽部
80 /*
90 /*     P R O G R A M E D   B Y   柏 木 勝 利
100 /*
110 /*
120 m_init()
130 /*
140 str  pd(30)[256]:pd(30)="[*]"
150 char po(255),v(4,10),vo(4,9)
160 /*
170 str p0="y3,0", p1="y3,1", p2="y3,2", p3="y3,3"
180 str bd="y2,23",sd="y2,17",hc="y2,65",ho="y2,66"
```

```
 190 str t1="y2,28",t2="y2,29",t3="y2,30",t4="y2,31"
 200 str c1="y2,3", cc="y2,5"
 210 /*
 220 for i=1 to 8
 230   m_alloc(i,3000)
 240   m_assign(i,i)
 250 next
 260 /*
 270 for i=1 to 8
 280   m_trk(i,"[d.c.][coda]")
 290 next
 300 /*
 310 VOI()
 320 MML1()
 330 MML2()
 340 m_play()
 350 end
 360 /*
 370 /*      T R A C K   S E T
 380 /*
 390 func trk(t)
 400   c=0
 410   while po(c)<>255
 420     m_trk(t,pd(po(c)))
 430     c=c+1
 440   endwhile
 450 endfunc
 460 /*
 470 /*      V O I C E   S E T
 480 /*
 490 func vset(no)
 500   v(0,0)=(vo(4,1)*8)+vo(4,0)
 510   v(0,1)=15
 520   v(0,9)=3
 530   for x=0 to 3
 540     for y=0 to 9
 550       v(x+1,y)=vo(x,y)
 560     next
 570   next
 580   m_vset(no,v)
 590 endfunc
 600 /*
 610 /*      G R A D I U S  I I I   V O I C E   D A T A
 620 /*
 630 func VOI()
 640 /*
 650 /*                                  "B A S S"
 660 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2
 670 vo={ 31,  11,   0,  10,  11,  34,   0,   0,   7,   0,
 680      31,  10,   1,  10,  12,  45,   0,   8,   5,   0,
 690      31,  10,   2,   9,  13,  32,   0,   0,   2,   0, /* C
ON  FBL
 700      31,   2,   0,  12,   8,   0,   0,   1,   3,   0,
 2,   7}
 710 vset(70)
 720 /*
 730 /*                                  "M E L O 1"
 740 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2
 750 vo={ 18,   8,   0,  10,   2,  29,   0,   4,   4,   0,
 760      28,   2,   1,  12,  15,   3,   0,   4,   4,   0,
 770      28,   2,   1,  12,  15,   3,   0,   4,   4,   0, /* C
ON  FBL
 780      28,   2,   1,  12,  15,   4,   0,   4,   4,   0,
 5,   7}
 790 vset(71)
 800 /*
 810 /*                                  "M E L O 2"
 820 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2
 830 vo={ 20,   4,   0,   8,   4,  29,   0,   4,   3,   0,
 840      20,   4,   0,  10,  12,   1,   0,   8,   3,   0,
 850      20,   4,   0,   8,   4,  14,   0,   4,   7,   0, /* C
ON  FBL
 860      20,   4,   0,  10,  12,   2,   0,   4,   7,   0,
 4,   5}
 870 vset(72)
 880 /*
 890 /*                                  "B R A S S"
 900 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2
 910 vo={ 20,   8,   0,  10,   2,  30,   0,   4,   0,   0,
 920      30,   2,   0,  12,  15,   4,   0,   4,   0,   0,
 930      30,   2,   0,  12,  15,   4,   0,   4,   0,   0, /* C
ON  FBL
 940      30,   2,   0,  12,  15,   4,   0,   4,   0,   0,
 5,   7}
 950 vset(73)
 960 /*
 970 /*                                  "G L O C K E N"
 980 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2
 990 vo={ 31,  24,   0,  12,  15,  35,   0,  14,   7,   0,
1000      31,  13,   0,   8,  15,   0,   0,   2,   7,   0,
1010      31,  20,   0,   4,  15,  36,   0,  13,   3,   0, /* C
ON  FBL
1020      31,  12,   0,   5,  15,   0,   0,   1,   3,   0,
 4,   3}
1030 vset(74)
1040 /*
1050 /*                                  "C H O R U S 1"
1060 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2
1070 vo={ 22,   0,   0,   0,   0,  24,   0,   4,   7,   0,
1080      20,   1,   0,   6,   2,   0,   0,   4,   7,   0,
1090      22,   0,   0,   0,   0,  45,   0,  12,   3,   0, /* C
ON  FBL
1100      20,   1,   0,   6,   2,   0,   0,   4,   3,   0,
 4,   7}
1110 vset(75)
1120 /*
1130 /*                                  "B - D R U M"
1140 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2
1150 vo={ 31,  28,   0,  15,  15,  19,   0,   1,   3,   2,
1160      31,  18,  11,  15,  15,   0,   0,   1,   7,   0,
1170      31,  18,  11,  12,  15,   0,   0,   1,   7,   0, /* C
ON  FBL
1180      31,  18,  11,  15,  15,   0,   0,   1,   3,   0,
 5,   6}
1190 vset(76)
1200 /*
1210 /*                                  "T I M P A N I"
1220 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2
1230 vo={ 31,   8,   0,   4,   2,   2,   0,  10,   2,   1,
1240      31,  18,   3,  15,  13,  42,   0,   0,   0,   2,
1250      31,  20,  12,  12,  14,  22,   0,   0,   3,   3, /* C
ON  FBL
1260      31,   6,   3,  15,  15,   0,   0,   1,   0,   0,
 3,   7}
1270 vset(77)
1280 /*
1290 /*                                  "R I D E"
1300 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2
1310 vo={ 31,  28,   0,   1,   5,  20,   0,   3,   3,   0,
1320      31,  16,   0,   8,   5,   4,   0,   7,   7,   1,
1330      31,  28,   0,   1,   5,  14,   0,   7,   7,   2, /* C
ON  FBL
1340      31,  16,   0,   8,   5,   0,   0,  11,   3,   3,
 4,   7}
1350 vset(78)
1360 /*
1370 /*                                  "S U B"
1380 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2
1390 vo={ 29,   1,   0,   2,   1,  26,   0,   1,   3,   0,
1400      31,   1,   0,  10,   2,   0,   0,   1,   3,   0,
1410      29,   1,   0,   2,   1,  32,   0,   1,   7,   0, /* C
ON  FBL
1420      31,   1,   0,  10,   2,   0,   0,   1,   7,   0,
 4,   7}
1430 vset(79)
1440 /*
1450 /*                                  "C H O R U S 2"
1460 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2
1470 vo={ 20,   0,   0,   0,   0,  24,   0,   4,   7,   0,
1480      15,   1,   0,   6,   2,   3,   0,   4,   7,   0,
1490      20,   0,   0,   0,   0,  45,   0,  12,   3,   0, /* C
ON  FBL
1500      15,   1,   0,   6,   2,   3,   0,   4,   3,   0,
 4,   7}
1510 vset(80)
1520 /*
1530 /*                                  "H I T"
1540 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2
1550 vo={ 28,   2,   0,   2,   1,  31,   0,   2,   3,   0,
1560      26,   1,   1,   6,   2,   6,   0,   6,   3,   0,
1570      28,   2,   0,   2,   1,  24,   0,   2,   7,   0, /* C
ON  FBL
1580      26,   1,   1,   6,   2,   6,   0,   6,   7,   0,
 4,   7}
1590 vset(81)
1600 /*
1610 endfunc
1620 /*
1630 /*      M M L   D A T A  -  1
1640 /*
1650 func MML1()
1660 m_tempo(161)
1670 /*
1680 /*｢ﾒﾛﾃﾞｨｰ｣
1690 pd(0)="@71@v122o2@l3q8p3y48,8"
1700 pd(1)="g4.&|:y48,193g-&y48,123g-&y49,193g-&y48,8g&y48,78g&
y48,148g&y48,78g&y48,8g&:|y48,183g-&y48,103g-&y48,183g-&y48,8g&y
48,88g&y48,168g&y48,88g&y48,8g@72g12b12<d12
1710 pd(2)="g4.&|:4y48,223g-&y48,183g-&y48,223g-&y48,8g&y48,48g
&y48,88g&y48,48g&y48,8g&:|y48,213g-&y48,163g-&y48,213g-&y48,8g&y
48,58g&y48,108g&y48,58g&y48,8g>@71
1720 pd(3)="a4.&|:y48,193a-&y48,123a-&y49,193a-&y48,8a&y48,78a&
y48,148a&y48,78a&y48,8a&:|y48,183a-&y48,103a-&y48,183a-&y48,8a&y
48,88a&y48,168a&y48,88a&y48,8a@72a12<c12f12
1730 pd(4)="a4.&|:4y48,223a-&y48,183a-&y48,223a-&y48,8a&y48,48a
&y48,88a&y48,48a&y48,8a&:|y48,213a-&y48,163a-&y48,213a-&y48,8a&y
48,58a&y48,108a&y48,58a&y48,8a>@71
1740 pd(5)="b4.&|:y48,193b-&y48,123b-&y49,193b-&y48,8b&y48,78b&
y48,148b&y48,78b&y48,8b&:|y48,183b-&y48,103b-&y48,183b-&y48,8b&y
48,88b&y48,168b&y48,88b&y48,8b@72b12<d12g12
1750 pd(6)="b4.&|:4y48,223b-&y48,183b-&y48,223b-&y48,8b&y48,48b
&y48,88b&y48,48b&y48,8b&:|y48,213b-&y48,163b-&y48,213b-&y48,8b&y
48,58b&y48,108b&y48,58b&y48,8b@71
1760 pd(7)="c4.&|:y48,193>b&y48,123b&y49,193b&y48,8<c&y48,78c&y
48,148c&y48,78c&y48,8c&:|y48,183>b&y48,103b&y48,183b&y48,8<c&y48
,88c&y48,168c&y48,88c&y48,8c@72c12f12a12<
1770 pd(8)="c4.&|:4y48,223>b&y48,183b&y48,223b&y48,8<c&y48,48c&
y48,88c&y48,48c&y48,8c&:|y48,213>b&y48,163b&y48,213b&y48,8<c&y48
,58c&y48,108c&y48,58c&y48,8c
1780 pd(9)="@71@v117o2g2<d2&|:3y48,193d-&y48,123d-&y48,193d-&y4
8,8d&y48,78d&y48,148d&y48,78d&y48,8d&:|y48,183d-&y48,103d-&y48,1
83d-&y48,8d&y48,88d&y48,168d&y48,88d&y48,8d
1790 pd(10)="d6c6>b6<c4&c6g12&g2&|:7y48,183g-&y48,103g-&y48,183
g-&y48,8g&y48,88g&y48,168g&y48,88g&y48,8g&:|y48,173g-&y48,83g-&y
48,173g-&y48,8g&y48,98g&y48,188g&y48,98g&y48,8g
1800 pd(11)="c6>b6<c6>a2&|:9y48,183a-&y48,103a-&y48,183a-&y48,8
a&y48,88a&y48,168a&y48,88a&y48,8a&:|y48,173a-&y48,83a-&y48,173@l
2a-<y48,8c6d6
1810 po={ 0,1,2,3,4,5,6,7,8,9,10,9,11,255 }
1820 trk(1)
1830 /*
1840 /*｢ﾒﾛﾃﾞｨｰ･ｴｺｰ｣
1850 pd(0)="@71@v110o2@l3q8p3y49,32r12"
1860 pd(1)="g4.&|:y49,217g-&y49,147g-&y49,217g-&y49,32g&y49,102
g&y49,172g&y49,102g&y49,32g&:|y49,207g-&y49,127g-&y49,207g-&y49,
32g&y49,112g&y49,192g&y49,112g&y49,32g@72g12b12<d12
```

```
 1870 pd(2)="g4.&|:4y49,247g-&y49,207g-&y49,247g-&y49,32g&y49,72
g&y49,112g&y49,72g&y49,32g&:|y49,237g-&y49,187g-&y49,237g-&y49,3
2g&y49,82g&y49,132g&y49,82g&y49,32g>@71
 1880 pd(3)="a4.&|:y49,217a-&y49,147a-&y49,217a-&y49,32a&y49,102
a&y49,172a&y49,102a&y49,32a&:|y49,207a-&y49,127a-&y49,207a-&y49,
32a&y49,112a&y49,192a&y49,112a&y49,32a@72a12<c12f12
 1890 pd(4)="a4.&|:4y49,247a-&y49,207a-&y49,247a-&y49,32a&y49,72
a&y49,112a&y49,72a&y49,32a&:|y49,237a-&y49,187a-&y49,237a-&y49,3
2a&y49,82a&y49,132a&y49,82a&y49,32a>@71
 1900 pd(5)="b4.&|:y49,217b-&y49,147b-&y49,217b-&y49,32b&y49,102
b&y49,172b&y49,102b&y49,32b&:|y49,207b-&y49,127b-&y49,207b-&y49,
32b&y49,112b&y49,192b&y49,112b&y49,32b@72b12<d12g12
 1910 pd(6)="b4.&|:4y49,247b-&y49,207b-&y49,247b-&y49,32b&y49,72
b&y49,112b&y49,72b&y49,32b&:|y49,237b-&y49,187b-&y49,237b-&y49,3
2b&y49,82b&y49,132b&y49,82b&y49,32b@71
 1920 pd(7)="c4.&|:y49,217>b&y49,147b&y49,217b&y49,32<c&y49,102c
&y49,172c&y49,102c&y49,32c&:|y49,207>b&y49,127b&y49,207b&y49,32<
c&y49,112c&y49,192c&y49,112c&y49,32c@72c12f12a12<
 1930 pd(8)="c4.&|:4y49,247>b&y49,207b&y49,247b&y49,32<c&y49,72c
&y49,112c&y49,72c&y49,32c&:|y49,237>b&y49,187b&y49,237b&y49,32<c
&y49,82c&y49,132c&y49,82c&y49,32c
 1940 pd(9)="@71@v105o2g2<d2&|:3y49,217d-&y49,147d-&y49,217d-&y4
9,32d&y49,102d&y49,172d&y49,102d&y49,8d&:|y49,207d-&y49,127d-&y4
9,207d-&y49,32d&y49,112d&y49,192d&y49,112d&y49,32d
 1950 pd(10)="d6c6>b6<c4&c6g12&g2&|:7y49,207g-&y49,127g-&y49,207
g-&y49,32g&y49,112g&y49,192g&y49,112g&y49,32g&:|y49,197g-&y49,10
7g-&y49,197g-&y49,32g&y49,122g&y49,202g&y49,122g&y49,32g
 1960 pd(11)="c6>b6<c6>a2&|:9y49,207a-&y49,127a-&y49,207a-&y49,3
2a&y49,112a&y49,192a&y49,112a&y49,32a&:|y49,197a-&y49,107a-&y49,
197@l2a-<y49,32c6d6
 1970 trk(2)
 1980 /*
 1990 /*｢ｸﾞﾛｯｹﾝ or ｽﾄﾘﾝｸﾞｽ｣
 2000 pd(0)="@74@v120o6l12q8p3y50,16"
 2010 pd(1)="|:4p1gp2d>p1bp2gp1b<p2d:||:4p1fp2c>p1ap2fp1a<p2c:|
 2020 pd(2)="|:4p1gp2d>p1bp2gp1b<p2d:||:p1fp2c>p1ap2fp1a<p2c:|
 2030 pd(3)="p1a&p3ap2f24g24p3ep1fp3dp2ep3cp1d>p3b<p2c>p3a
 2040 pd(4)="@79@v111o5|:4gbg<d>b<c>:||:4faf<c>ab:|
 2050 pd(5)="|:4gbg<d>b<c>:||:faf<c>ab:|"+pd(0)+"@v122o4a<cfcfaf
a<c>a<cf
 2060 pd(6)="@v120|:p1bp2gp1ep2cp1ep2g:||:p1ap2f+p1d>p2b<p1dp2f+
:|
 2070 pd(7)="|:p1gp2ep1c>p2a<p1cp2e:|>>p1gp3ap2b<p3cp1dp3ep2f+p3
gp1ap3b<p2cp3d
 2080 pd(8)="|:p1gp2e->p1b-p2gp1b-<p2e-:||:p1ap2fp1c>p2a<p1cp2f:
|
 2090 pd(9)="p1bp2gp1dp2gp1d>p2b<p1ap2fp1cp2fp1c>p2a<
 2100 pd(10)="p1gp2e>p1b<p2e>p1bp2g<p1ep2c>p1g<p2c>p1gp2e
 2110 pd(11)="|:p1gp2b<p1d>p2gp1b<p2g>:||:p1a<p2dp1f+>p2a<p1dp2a
>:|
 2120 pd(12)="|:p1b<p2ep1g>p2b<p1ep2b>:|<p1cp2ep1gp2cp1e<p2c>>p1
a<p2dp1f+>p2a<p1dp2a>
 2130 po={ 0,1,2,3,4,5,6,7,6,8,9,10,11,12,11,12,30,255 }
 2140 trk(3)
 2150 /*
 2160 /*｢ﾌﾞﾗｽ1｣
 2170 pd(0)="@73@v108o2l1q8p3y51,12"
 2180 pd(1)="|:@76@v124a8@73@v108g2..&g1@76@v124a8@73@v108f2..&f
1:|
 2190 pd(2)="@v104l6p1b1b2b<cd>a2.a12a12a12a2agab1&b2b<cd>f1&f1
 2200 pd(3)="@v108o3l1p3edcdede-f
 2210 pd(4)="@76@v122>a8<@73@v108g4.@76@v122>a8<@73@v108f4.@76@v
122>a8<@73@v108e4.c2|:dd>bg2a2<:|
 2220 po={ 0,1,2,3,4,30,255 }
 2230 trk(4)
 2240 /*
 2250 /*｢ﾌﾞﾗｽ2｣
 2260 pd(0)="@73@v107o2l1q8p3y52,16"
 2270 pd(1)="|:d&dc&c:|
 2280 pd(2)="@v103l6p3g1g2gabf2.f12f12f12f2fdfg1&g2gabc1&c1
 2290 pd(3)="@v107o2l1bagabab-<c
 2300 pd(4)="d2c2>b2g2|:gf+ec2d2:|
 2310 trk(5)
 2320 /*
 2330 /*｢ｺｰﾗｽ1｣
 2340 pd(0)="@75o2l4q8p3y53,20"
 2350 pd(2)="|:@v125g&@v124g&@v123g&@v122g<@v127g&@v125g&@v123g&
@v121g>
 2360 pd(3)="@v125f&@v124f&@v123f&@v122f<@v127f&@v125f&@v123f&@v
121f>:|
 2370 pd(4)="@73@v104l6p2y53,32b1b2b<cd>a2.a12a12a12a2agab1&b2b<
cd>f1
 2380 pd(5)="@74@v114o4l12rp1a<p2cp1fp2cp1fp2ap1fp2a<p1c>p2a<p1c
 2390 pd(6)="@80@v127o3l4p3y53,20"
 2400 pd(7)="@v127e&@v124e&@v121e&@v118e":pd(8)="@v127d&@v124d&@
v121d&@v118d
 2410 pd(9)="@v127c&@v124c&@v121c&@v118c":pd(10)="@v127e-&@v124e
-&@v121e-&@v118e-
 2420 pd(11)="@v127f&@v124f&@v121f&@v118f
 2430 pd(12)=pd(7)+pd(8)+pd(9)+pd(8)+pd(7)+pd(8)+pd(10)+pd(11)
 2440 pd(13)="@81@v123o4g2f2e2d2
 2450 pd(14)=pd(6)+">|:@v126g&@v123g&@v120g&@v117g<@v126d&@v123d
&@v120d&@v117d
 2460 pd(15)="@v126e&@v123e&@v120e&@v117e@v126c&@v122c>|1@v126a&
@v122a:||2<@v126d&@v122d
 2470 po={ 0,2,3,4,5,6,12,13,14,15,30,255 }
 2480 trk(6)
 2490 /*
 2500 /*｢ｺｰﾗｽ2｣
 2510 pd(0)="@75o2l4q8p3y54,12":pd(1)="@77@v127o3@l2e&e-&d&d-&c&
>b&b-&a"+pd(0)
 2520 pd(2)="|:@v118d&@v117d&@v116d&@v115d<@v119d&@v117d&@v115d"
+pd(1)+pd(1)+pd(1)
 2530 pd(3)="@v118c&@v117c&@v116c&@v115c<@v119c&@v117c&@v115c&@v
113c:|
 2540 pd(4)="@78@v120o4l12y54,20|:7c6cccc6c6c6c:|c6c6c6r2
 2550 pd(6)="@80@v123o2l4p3y54,12"
 2560 pd(7)="@v123b&@v120b&@v117b&@v114b":pd(8)="@v123a&@v120a&@
v117a&@v114a
 2570 pd(9)="@v123g&@v120g&@v117g&@v114g":pd(10)="@v123b-&@v120b
-&@v117b-&@v114b-
 2580 pd(11)="o3@v123c&@v120c&@v117c&@v114c
 2590 pd(12)=pd(7)+pd(8)+pd(9)+pd(8)+pd(7)+pd(8)+pd(10)+pd(11)
 2600 pd(13)="@81@v115o4y54,32r12g2f2e2@l80d
 2610 pd(14)=pd(6)+"|:@v122d&@v119d&@v116d&@v113d@v122a&@v119a&@
v116a&@v113a
 2620 pd(15)="@v122b&@v119b&@v116b&@v113b@v122g&@v118g|1@v122d&@
v118d:||2@v122a&@v118a
 2630 po={ 0,2,3,4,6,12,13,14,15,30,255 }
 2640 trk(7)
 2650 /*
 2660 /*｢ﾍﾞｰｽ & ﾄﾞﾗﾑ｣
 2670 pd(0)="@70@v127o2l12q8p3y15,0y55,0"+p3
 2680 pd(1)=cl+"g6g"+sd+"g"+hc+"g|:"+bd+"g:|"+hc+"g"+bd+"g"+sd+"
g"+hc+"<d>"+bd+"g
 2690 pd(2)=bd+"g&"+hc+"g"+bd+"g"+sd+"g"+hc+"g|:"+bd+"g:|"+hc+"g
"+bd+"g"+sd+"g"+sd+"<d>"+sd+"g
 2700 pd(3)=bd+"g&"+hc+"g"+bd+"g"+sd+"g"+hc+"g|:"+bd+"g:|"+hc+"g
"+bd+"g"+sd+"g"+hc+"<d>"+bd+"g
 2710 pd(4)="|:"+t1+"g24&:||:"+t1+"g:||:3"+t2+"g:||:3"+t3+"g:|"+
t4+"g"+t4+"<d>"+t4+"g
 2720 pd(5)="|:7"+bd+"g&"+hc+"g"+bd+"g"+sd+"g"+hc+"<d>"+bd+"g"+b
d+"g6"+bd+"g"+sd+"g"+bd+"<d>"+bd+"g:|"+bd+"g&"+hc+"g"+bd+"g"+sd+
"g"+hc+"<d>"+bd+"g"+sd+"g&|:3"+sd+"g:|"+sd+"<d>"+sd+"g<
 2730 pd(6)=bd+"c6"+bd+"c"+sd+"c|:"+bd+"c:|"+bd+"c24&"+bd+"c24"+
bd+"c24&"+bd+"c24@l4c&l12"+bd+"c16"+sd+"cgc>
 2740 pd(7)=bd+"b6"+bd+"b"+sd+"b|:"+bd+"b:|"+bd+"b24&"+bd+"b24"+
bd+"b24&"+bd+"b24@l4b&l12"+bd+"b16"+sd+"b<f+>b
 2750 pd(8)=bd+"a6"+bd+"a"+sd+"a|:"+bd+"a:|"+bd+"a24&"+bd+"a24"+
bd+"a24&"+bd+"a24@l4a&l12"+bd+"a16"+sd+"a<e>a<
 2760 pd(9)="|:"+t1+"d24&:|"+t1+"d"+t1+"d|:3"+t2+"d:|"+t3+"d"+t3
+"c"+t3+">b<"+t4+"c"+t4+">b"+t4+"a<
 2770 pd(10)=bd+"o3e-6"+bd+"e-"+sd+"e-|:"+bd+"e-:|"+bd+"e-24&"+b
d+"e-24"+bd+"e-24&"+bd+"e-24@l4e-&l12"+bd+"e-16"+sd+"e-b-e-
 2780 pd(11)=bd+"f6"+bd+"f"+sd+"f|:"+bd+"f:||:3"+bd+"f24&:|"+bd+
"f24@l4f&l12"+bd+"f16"+sd+"f"+sd+"e"+sd+"d
 2790 pd(12)=cl+"g4"+sd+">g<"+sd+"g"+sd+">g<"+cl+"f4"+sd+">f<"+s
d+"f"+sd+">f<
 2800 pd(13)=cl+"e4"+sd+">e<"+sd+"e"+sd+">e<"+t1+"d"+t2+"c"+t2+"
>b"+t3+"a"+t3+"g"+t4+"f+
 2810 pd(14)=cl+"g6g"+sd+"g"+hc+"g|:"+bd+"g:|"+hc+"<d>"+bd+"g"+s
d+"g"+hc+"b"+bd+"<d>
 2820 pd(15)=bd+"f+&"+hc+"f+"+bd+"f+"+sd+"f+"+hc+"f+|:"+bd+"f+:|
"+hc+"<d>"+bd+"f+"+sd+"f+"+hc+"a"+bd+"<d>
 2830 pd(16)=bd+"e&"+hc+"e"+bd+"e"+sd+"e"+hc+"e|:"+bd+"e:|"+hc+"
b"+bd+"e"+sd+"e"+hc+"g"+bd+"b<
 2840 pd(17)=bd+"c&"+hc+"c"+bd+"c"+sd+"c"+hc+"c"+bd+"c"+t1+"d&"+
t1+"d"+t2+"d"+t2+"d"+t3+"c"+t4+">b
 2850 pd(18)=bd+"g&"+hc+"g"+bd+"g"+sd+"g"+hc+"g|:"+bd+"g:|"+hc+"
<d>"+bd+"g"+sd+"g"+hc+"b"+bd+"<d>
 2860 pd(19)=bd+"e&"+hc+"e"+bd+"e"+sd+"e"+hc+"e"+bd+"e"+sd+"e"+s
d+"b|:"+sd+"e:|"+sd+"g"+sd+"b<
 2870 pd(20)=sd+"c&"+bd+"c"+sd+"c"+bd+"c"+sd+"c"+bd+"c|:"+t1+"d2
4&:|"+t2+"d"+t2+"d"+t3+"d"+t3+"c"+t4+">b
 2880 po={ 0,1,2,1,3,1,2,1,4,5,6,7,8,9,6,7,10,11,12,13,14,15,16,
17,18,15,19,20,30,255 }
 2890 trk(8)
 2900 /*
 2910 endfunc
 2920 /*
 2930 /*
 2940 func MML2()
 2950 /*
 2960 /*｢ﾒﾛﾃﾞｨｰ｣
 2970 pd(0)="@71@v119o3@l3q8p3y48,8"
 2980 pd(1)="e4.&y48,203e-&y48,143e-&y48,203e-&y48,8e&y48,68e&y4
8,128e&y48,68e&y48,8ee6f+6g6
 2990 pd(2)="d4.&|:y48,203d-&y48,143d-&y48,203d-&y48,8d&y48,68d&
y48,128d&y48,68d&y48,8d&:|y48,193d-&y48,123d-&y48,193@l2d-y48,8@
l3c6>b6
 3000 pd(3)="a4.&|:y48,203a-&y48,143a-&y48,203a-&y48,8a&y48,68a&
y48,128a&y48,68a&y48,8a&:|y48,193a-&y48,123a-&y48,193a-&y48,8a&y
48,78a&y48,148a&y48,78a&y48,8ag4l12gab<cdef+gab<cd@l3>
 3010 pd(4)="d4.&|:y48,203d-&y48,143d-&y48,203d-&y48,8d&y48,68d&
y48,128d&y48,68d&y48,8d&:|y48,193d-&y48,123d-&y48,193d-&y48,8d&y
48,78d&y48,148d&y48,78d&y48,8df+4
 3020 pd(5)="g4.&y48,203g-&y48,143g-&y48,203g-&y48,8g&y48,68g&y4
8,128g&y48,68g&y48,8gg6a6b-6a6b-6<c6&c4c12>b12<c12
 3030 pd(6)="@v120d4.&|:12y48,183d-&y48,103d-&y48,183d-&y48,8d&y
48,88d&y48,168d&y48,88d&y48,8d&:|y48,173d-&y48,83d-&y48,173d-&y4
8,8d&y48,98d&y48,188d&y48,98d&y48,8d>>
 3040 pd(7)="@v119g4.&|:y48,193g-&y48,123g-&y48,193g-&y48,8g&y48
,78g&y48,148g&y48,78g&y48,8g&:|y48,183g-&y48,103g-&y48,183g-&y48
,8g&y48,88g&y48,168g&y48,88g&y48,8gg12a12g12<
 3050 pd(8)="d4.&|:y48,193d-&y48,123d-&y48,193d-&y48,8d&y48,78d&
y48,148d&y48,78d&y48,8d&:|y48,183d-&y48,103d-&y48,183@l2d-y48,8@
l3c6>b6
 3060 pd(9)="g4.&y48,193g-&y48,123g-&y48,193g-&y48,8g&y48,78g&y4
8,148g&y48,78g&y48,8g":pd(10)=pd(9)+"f+6d6>a6":pd(11)=pd(9)+"g6<
e6g6
 3070 pd(12)="f+6g6a4.&|:y48,193a-&y48,123a-&y48,193a-&y48,8a&y4
8,78a&y48,148a&y48,78a&y48,8a&:|y48,183a-&y48,103a-&y48,183@l2a-
 3080 po={ 0,1,2,3,1,4,5,6,7,8,7,10,7,8,11,12,30,255 }
 3090 trk(1)
 3100 /*
 3110 /*｢ﾒﾛﾃﾞｨｰ･ｴｺｰ｣
 3120 pd(0)="@71@v107o3@l3q8p3y49,32"
 3130 pd(1)="e4.&y49,227e-&y49,167e-&y49,227e-&y49,32e&y49,92e&y
49,152e&y49,92e&y49,32ee6f+6g6
 3140 pd(2)="d4.&|:y49,227d-&y49,167d-&y49,227d-&y49,32d&y49,92d
&y49,152d&y49,92d&y49,32d&:|y49,217d-&y49,147d-&y49,217@l2d-y49,
32@l3c6>b6
 3150 pd(3)="a4.&|:y49,227a-&y49,167a-&y49,227a-&y49,32a&y49,92a
&y49,152a&y49,92a&y49,32a&:|y49,217a-&y49,147a-&y49,217a-&y49,32
a&y49,102a&y49,172a&y49,102a&y49,32ag4l12gab<cdef+gab<cd@l3>
 3160 pd(4)="d4.&|:y49,227d-&y49,167d-&y49,227d-&y49,32d&y49,92d
&y49,152d&y49,92d&y49,32d&:|y49,217d-&y49,147d-&y49,217d-&y49,32
d&y49,102d&y49,172d&y49,102d&y49,32df+4
```


▶最近のCDプレイヤーには演奏スピードが±10%まで変えることができるものがありますが，岡村孝子を５％増しのスピードで聴くとアイドルバージョンになります。これがなかなかよい。お試しあれ。
佐藤　昌幸(23)東京都

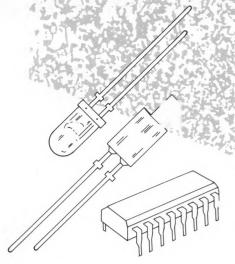
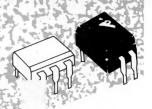
```
 3170 pd(5)="g4.&y49,227g-&y49,167g-&y49,227g-&y49,32g&y49,92g&y
49,152g&y49,92g&y49,32gg6a6b-6a6b-6<c6&c4c12>b12<c12
 3180 pd(6)="@v110d4.&|:12y49,207d-&y49,127d-&y49,207d-&y49,32d&
y49,112d&y49,192d&y49,112d&y49,32d&:|y49,197d-&y49,107d-&y49,197
d-&y49,32d&y49,122d&y49,212d&y49,122d&y49,32d>>
 3190 pd(7)="@v107g4.&|:y49,217g-&y49,147g-&y49,217g-&y49,32g&y4
9,102g&y49,172g&y49,102g&y49,32g&:|y49,207g-&y49,127g-&y49,207g-
&y49,32g&y49,112g&y49,192g&y49,112g&y49,32gg12a12g12<
 3200 pd(8)="d4.&|:y49,217d-&y49,147d-&y49,217d-&y49,32d&y49,102
d&y49,172d&y49,102d&y49,32d&:|y49,207d-&y49,127d-&y49,207@12d-y4
9,32@13c6>b6
 3210 pd(9)="g4.&y49,217g-&y49,147g-&y49,217g-&y49,32g&y49,102g&
y49,172g&y49,102g&y49,32g":pd(10)=pd(9)+"f+6d6>a6":pd(11)=pd(9)+
"g6<e6g6
 3220 pd(12)="f+6g6a4.&y49,217a-&y49,147a-&y49,217a-&y49,32a&y49
,102a&y49,172a&y49,102a&y49,32a&y49,207a-&y49,127a-&y49,207a-&y4
9,32a&y49,102a&y49,172@11a
 3230 trk(2)
 3240 /*
 3250 endfunc
 3260 /*
 3270 /*    ご゛くろうさまで゛した。
 3280 /*
```

リスト2　メタルサイト

```
10 '■||▮▮▮▮■■■■■■■■■■■■■▮▮▮▮||■
20 '■|  MISSION METAL SIGHT    |■
30 '■|     - STAGE 01 -        |■
40 '■|(C) Team CROSS WONDER    |■
50 '■|   Arranged BY TECCHAN   |■
60 '■|   1990.9.1-1990.9.12    |■
70 '■|    Thanks to Yonasan    |■
80 '■||▮▮▮▮■■■■■■■■■■■■■▮▮▮▮||■
90 DEFSTR A-H:TEMPO 0:"INST"
100 PLAY "T130";
110 '||▮▮▮■■ BASS ■■▮▮▮||
120 A="V6F+C+EF+C+EF+C+V8EF+C+EF+V9C+EF+C+EF+C+EF+C+EV10F+C+EF+G
+AG+E
130 B="V12F+C+EF+C+EF+C+EF+C+EF+V14C+EF+C+EF+C+EF+C+EF+C+EF+G+AG
+E
140 C=STRING$(8,"F+C+E")+"F+C+EF+G+AG+E" '1
150 D=STRING$(8,"F+C+E")+"F+C+EF+GAGE"
160 PLAY "O3i6L16"+A+B;
170 PLAY C+C;
180  PLAY "["+STRING$(4,C);
190 FOR Z=0 TO 1
200 PLAY STRING$(3,C);
210 PLAY STRING$(3,C);
220 PLAY STRING$(3,C);
230 PLAY STRING$(2,C);
240 NEXT
250 PLAY C+D+"]";
260 '||▮▮▮■■ CHORD ■■▮▮▮||
270 A="O4i2B8.B16R8B8&B2"
280 A1="O4i2F+8.F+16R8F+8&F+2"
290 A2="O4i2C+8.C+16R8C+8&C+2"
300 B= "V14i4O5C+8.<B16R8>C+8&C+2&C+2.R8C+16R16<B1&B1>"
310 B1="V14i4O4G+8.G+16R8 G+8&G+2&G+2.R8G+16R16F+1&F+1"
320 B2="V14i4O4 E8. E16R8  E8& E2& E2.R8 E16R16C+1&C+1"
330 D= "C+8.<B16R8>C+8&C+2&C+2.R8C+16R16<B8.B16R8B8&B2&B1"
340 D1="G+8.G+16R8 G+8&G+2&G+2.R8G+16R16 F+8.F+16R8F+8&F+2&F+1"
350 D2="E8.E16R8E8&E2&E2.R8E16R16C+8.C+16R8C+8&C+2&C+1"
360 E= "A8.A16R8A8&A2&A1B8.B16R8B8&B2&B1>C+8.C+16R8C+8&C+2&C+1"
370 E1="F+8.F+16R8F+8&F+2&F+1E8.E16R8E8&E2&E1F+8.F+16R8F+8&F+2&F
+1"
380 E2=STRING$(3,"C+8.C+16R8C+8&C+2&C+1")
390 PLAY ":R1R1R1R1V6P1"+A+"V8 "+A+"V10"+A+"V14"+A;
400 PLAY "["+A+A+A+A;
410 PLAY "R1R1R1R1";
420 FOR Z=0 TO 1
430 PLAY B+B+B;
440 PLAY D;
450 PLAY E;
460 NEXT
470 PLAY A+A+A+A+"]";
480 PLAY ":R1R1R1R1V6P2"+A1+"V8 "+A1+"V10"+A1+"V13"+A1;
490 PLAY "["+A1+A1+A1+A1;
500 PLAY "R1R1R1R1";
510 FOR Z=0 TO 1
520 PLAY B1+B1+B1;
530 PLAY D1;
540 PLAY E1;
550 NEXT
560 PLAY A1+A1+A1+A1+"]";
570 PLAY ":R1R1R1R1V6P3"+A2+"V8 "+A2+"V10"+A2+"V13"+A2;
580 PLAY "["+A2+A2+A2+A2;
590 PLAY "R1R1R1R1";
600 FOR Z=0 TO 1
610 PLAY B2+B2+B2;
620 PLAY D2;
630 PLAY E2;
640 NEXT
650 PLAY A2+A2+A2+A2+"]";
660 '||▮▮▮■■ BASS 2 ■■▮▮▮||
670 A=":O3i7V15P3L16"
680 B="F+8>F+<F+&F+F+>F+<F+F+F+>F+8E8C+8<"
690 C="F+8>F+<F+&F+F+>F+<F+F+F+>F+<F+>B>C+<BE<"
700 D="F+8>F+<F+F+2>B>C+<BE<"
710 E="B8>B<B&BB>B<BBB>A8G+8E8<A8>A<A&AA>A<AAA>F+8B>C+<BE<"
720 F="G+8>G+<G+&G+G+>G+<G+G+G+>A8G+8E8<G8>G<G&GG>G<GGG>F+8B>C+<
BE<"
730 PLAY A+STRING$(8,"R1");
740 PLAY "["+B+C+B+C;
750 PLAY B+C+B+D;
760 FOR Z=0 TO 1
770 PLAY B+C+B+C;
780 PLAY B+C+B+C;
790 PLAY B+C+B+C;
800 PLAY B+C+B+C;
810 PLAY B+C+B+C+B+C;
820 NEXT
830 PLAY E+F+"]";
840 '||▮▮▮■■ Drum ■■▮▮▮||
850 A="V15P3R1L8O3i39R2E32E32E16E16R16E16E16E16E16"
860 B=STRING$(4,"O1i40CO3i39E")
870 C=STRING$(3,"O1i40CO3i39E")+"E32E32E16E16E16"
880 D=STRING$(2,"O1i40CO3i39E")+"R4E16E16E16E16"
890 E=STRING$(2,"O1i40Ci39O3E")+"E16E16E16E16E16E16E8"
900 PLAY ":R1R1R1R1R1R1";
910 PLAY A;
920 PLAY "["+STRING$(4,B);
930 PLAY STRING$(3,B);
940 PLAY C;
950 FOR Z=0 TO 1
960 PLAY B+B+B+B;
970 PLAY B+B+B+D;
980 PLAY B+B+B+B;
990 PLAY B+B+B+B;
1000 PLAY B+B+B+B+B;
1010 PLAY D;
1020 NEXT
1030 PLAY D+D+D+E+"]";
1040 '||▮▮▮■■ MELODY ■■▮▮▮||
1050 A="O2i1V11"+STRING$(16,"F+1&")
1060 B="i3R2R8L8C+EF+G+F+16E16&E2."
1070 C="R4R8C+EF+G+F+G+8.+A8.B&B2"
1080 D="R2R8C+EF+G+A16G+16&G+2EC+"
1090 E="<B8.>C+8.+F+8&F+2&F+1"
1100 F="F+2.G+AB8.>C+8.<F+8&F+4.E"
1110 G="F+2.G+AB8.>C+8.F+8&F+2<"
1120 H="O2i1V11"+STRING$(12,"F+1&")
1130 PLAY ":O2P1"+A+"[";
1140 FOR Z=0 TO 1
1150 PLAY "O6V16"+B+C+D+E;
1160 PLAY "O6"+B+C+D+F+F+F+G;
1170 NEXT
1180 PLAY H+"]";
1190 '||▮▮▮■■ MELODY_SUB ■■▮▮▮||
1200 PLAY ":O2i1K20P2R@3"+A+"[";
1210 FOR Z=0 TO 1
1220 PLAY "K10O6V16"+B+C+D+E;
1230 PLAY "O6"+B+C+D+F+F+F+G;
1240 NEXT
1250 PLAY H+"]";
1260 PLAY ":O2K5Y7,28"+A+"[";
1270 FOR Z=0 TO 1
1280 PLAY "K3O3V11"+B+C+D+E;
1290 PLAY B+C+D+F+F+F+G;
1300 NEXT
1310 PLAY H+"]";
1320 PLAY ":O2K10R@3"+A+"[";
1330 FOR Z=0 TO 1
1340 PLAY "K0O3V11"+B+C+D+E;
1350 PLAY B+C+D+F+F+F+G;
1360 NEXT
1370 PLAY H+"]";
1380 '||▮▮▮■■ HI-HAT ■■▮▮▮||
1390 A="V12O2L16S0,0,0,0=1^1y6,30Q5"
1400 B="C8C8CCCCCCy6,12Cy6,30CC8C8"
1410 C="C8C8CCCCCCy6,12Cy6,30CR4"
1420 D="O3Q8y6,5C8Q5y6,30O2C8CCCCCCy6,12Cy6,30CC8C8"
1430 PLAY ":"+STRING$(8,"R1");
1440 PLAY "["+A+B+B+B+C;
1450 PLAY D+B+B+C;
1460 FOR Z=0 TO 1
1470 PLAY STRING$(7,B)+C;
1480 PLAY D+STRING$(6,B);
1490 PLAY STRING$(6,B)+C;
1500 NEXT
1510 PLAY B+B+B+"C8C8CCCCR2"+"]"
1520 END
1530 LABEL "INST"
1540 MEM$(&HB190,36)=HEXCHR$("F5 00 31 22 40 62 14 1B 1B 11 14 1
4 14 14 00 05 00 00 00 01 00 00 51 A8 F8 F5 00 00 00 00 00 00 80
 00 02 00") 'I1 MELODY_1
1550 MEM$(&HB1B4,36)=HEXCHR$("F3 00 45 42 41 42 16 1B 1B 0C 9C 1
C 1C 18 02 02 00 08 00 04 06 04 04 04 04 F6 00 00 80 00 00 CC 80
 00 02 00") 'I2 CHORD_1
1560 MEM$(&HB1D8,36)=HEXCHR$("DA 50 20 32 31 70 27 1A 23 00 DE D
F 5E 4D 01 01 01 82 00 00 00 00 11 15 15 04 00 88 80 80 0C CD 8A
 00 02 00") 'I3 MELODY_2
1570 MEM$(&HB1FC,36)=HEXCHR$("F8 00 02 01 00 00 25 1E 1B 07 94 5
4 54 54 01 06 06 06 07 06 06 08 A4 A4 14 F6 00 00 00 00 00 EC B2
 00 02 00") 'I4 CHORD_2
1580 MEM$(&HB244,36)=HEXCHR$("20 00 66 65 60 61 1C 3A 16 00 DF D
F 9F 9F 06 05 08 05 06 05 05 07 24 14 14 F4 00 00 00 00 00 00 00
 00 00 00") 'I6  BASS 1
1590 MEM$(&HB268,36)=HEXCHR$("F8 00 36 35 30 31 1D 2F 1D 00 DF D
F 9A 9F 07 06 09 88 07 06 06 04 29 19 19 39 80 00 00 00 00 C8 80
 00 02 00") 'I7  BASS 2
1600 MEM$(&HB6E8,36)=HEXCHR$("FC 00 0C 01 00 00 00 07 11 07 1F 9
C 5E 5C 00 0D 11 0D C0 00 80 00 01 F7 D9 F7 00 00 00 00 F4 C8 80
 00 02 00")  'I39 SnareDrum2
1610 MEM$(&HB70C,36)=HEXCHR$("80 00 01 0E 00 00 00 00 07 00 1E 1
E 1E 1D 1A 1C 0F 8C 40 C0 40 00 FD FE F8 F8 00 00 00 00 D0 C8 80
 00 02 00")  'I40 Bass Drum1
1620 RETURN
```

▶SX-WINDOWのワンテク。プロセス情報のウィンドウの中をクリックすると「kバイト」がバイト表示になる。　木全　和茂(17)愛知県

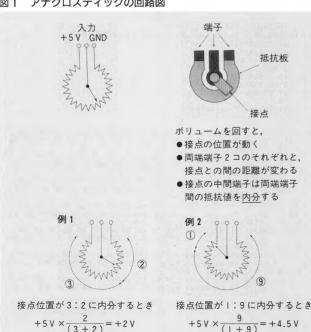
ハードウェア工作入門《6》

Misawa Kazuhiko
三沢 和彦

スティックを使ってブロック崩しを作ってみましょう。プログラムはX-BASICで書いたものをコンパイルして使用します。

先月までに，簡単な回路で応用性抜群のA/Dコンバータボードを製作し，まず最初の応用として簡易アナログジョイスティックを紹介してみました。今回は，アナログジョイスティックを使った簡単なゲームプログラムを組んでみましょう。内容は，昔なつかしの「ブロック崩し」です。

アナログジョイスティックの使い方

まずは，A/Dコンバータとボリューム1個の組み合わせでアナログジョイスティックとして活用する方法を考えてみましょう。図1が簡易アナログジョイスティックの全回路図です。

ボリュームには1本の抵抗の両端にある端子2個とその抵抗の途中に接触している中間端子1個とがあります。ボリュームの固定抵抗の両端に本体電源からの＋５VとGNDがつながっていて，ボリュームの中間端子がA/Dコンバータの電圧入力端子につながっています。

この中間端子からは，ボリュームの両端の電圧以下の電圧を連続的に取り出すことができます。それは，この中間端子がボリュームの抵抗線にじかに接触していて，ボリュームを回すことによって，両端の固定端子からの距離を連続的に変化させることができるからです。そして，両端のそれぞれの端子までの長さにしたがって内分された電圧が取り出されることになります（図１参照)。

したがって，ボリュームのツマミの位置によって中間端子の電圧が変わることになります。先月掲載のサンプルプログラムでは画面上に現れる数字がその中間端子の電圧を示していて，(ただしフルスケール＋５Vを256としたときの0～255の256段階）ツマミを回すとその数字も変化するのです。ツマミを左に一杯に回すと０，右に一杯に回すと255（誤差により最小で１か２，最大で253ぐらいのときもある）になるのがわかるでしょう。

図１　アナグロスティックの回路図

そこで，このようにして読み出された値をゲーム画面の上で動くプレイヤーの位置に対応させるのです。今月のプログラムリストの1020行以降が，そのルーチンになっています。これは単に読み出してきたデータをX座標としてパドルを定義したスプライトを移動させているだけです。もちろんA/Dコンバータからの読み出しは先月のルーチンとまったく同じものです。

このように，アナログジョイスティックとはいっても実は256段階のデジタル変化ですが，コンパイル後のプログラムでは，画面上のパドルの動きはずいぶんスムーズであることが実感できると思います。

ブロック崩しプログラミング

ゲームのプログラミング自体は簡単で，むしろ皆さんのほうがデザインセンスが優れていると思います。今回のサンプルゲームは，自作の簡易アナログジョイスティックでパドルを左右に動かし，ボールを返しながら，ブロックを消していきます。１ステージをクリアするとブロックの数が増え，パドルが前に進んでいきます。プレイヤーは持ち数が３で，これも１ステージクリアごとにひとつずつ増えていきます。

ボールとブロックとの衝突判断も若干不自然なところが残っていますし，ステージをクリアし続けるとパドルがブロックよりも前に出てしまうバグがありますが，X-BASICでの改良は皆さんにお任せします。

ただし，コンパイルしないとまったくゲームになりません。コンパイルは，

CC GAME.BAS IODRV.O

とすれば，実行ファイルGAME.Xが生成されます。なお，XCのver.2.0を使った場合には，コンパイルの途中で，

ERROR 15： function return value mismatch

ERROR 16： argument type mism

atch

の2種類の警告（Warning）が出ますが，これは無視してかまいません。

A/Dコンバータ用外部関数

このA/Dコンバータは次回のセンサ応用編でも使います。それだけ応用範囲の広い外部機器なのです。X68000からコントロールするときはread関数，start関数，clock関数をまったく同じ形で使えるのですが，これだけ汎用性があるのですから，その都度X-BASICのプログラムのなかで定義しないで，このread関数もマシン語で外部関数にしてしまうと便利です。

というわけで，おなじみのioinp関数，ioout関数に加えて，adread関数を新しく付け加えました。リスト2は以前載せた外部関数のソースリストにそのまま加えたもので，ioinp関数，ioout関数の部分はまったく変更ありませんので，以前と同様に使えます。付け加えたadread関数の部分は，

＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊

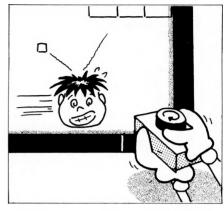
（追加した部分）

＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊

のように囲ってあります。このソースファイルを新たに，

CC IODRV.S

でコンパイルし，出来上がったIODRV.XをIODRV.FNCにリネームしてやれば，BASICでのread関数とまったく同じ使い方ができるようになります（readをadreadに書き直すだけ！）。

この外部関数を使えば，次回のプログラミングはたいへん短いものになりますが，どうもアセンブラは苦手で最初のiodrv.fncだけでも大変だったという初心者の人は，今までどおりBASICのread関数を使っていてもかまいません。

以上，自作A/Dコンバータ基板の簡単な応用を紹介してみました。次回からは，より身近で，より実用的な回路に挑戦してみたいと思いますので，お楽しみに。

リスト1　ブロック崩し

```
 10 /*  save "d:¥basic¥game.bas
 20 /*  save@"d:¥basic¥game.doc
 30 /**********************************************************
 40 /*
 50 /*  A／Dコンバータ＆簡易ジョイスティック
 60 /*       サンプルプログラム
 70 /*
 80 /*     なつかしのブロック崩し
 90 /*
100 /*     1990.10.10.  K.Misawa
110 /*
120 /**********************************************************
130 screen 1,1,1,1
140 vpage(7)
150 console,,0 : color 6
160 int v=0,cs=0,di=0,sgl,sign
170 int ch=0
180 int bx,by,sx=0,sy,px,x,xx,spd=2
190 int rwall=250,lwall=10,uwall=10,bdiam=8
200 int bhight=8,bwidth=16,braw=1,bcol=15
210 int ofs,ppos=466,base=32
220   ofs=bwidth/2 : bhight=4*spd
230 int col,raw,colnxt,rawnxt,pl
240 int stage=0,count=0,total=0,maxplayer=3,player
250   player=maxplayer-1
260 int flg,endflg=1,hscore=0
270 /*
280 dim int block(17,50)
290 dim int cl(5)= {3,5,7,9,11,13}
300 dim float gpat(375)
310 locate 42,6: print "HIGH SCORE"
320 locate 42,8: print "         0"
330 locate 42,12: print "     SCORE"
340 locate 42,18: print "     STAGE"
350 locate 42,24: print "    PLAYER"
360 /*
370 sp_init()
380 sp_off(0,127)
390 spdef()
400 sp_disp(1)
410 wall()
420 back_ground()
430 /*
440 /*メインルーチン
450 /*
460 while endflg
470   initialize()
480   stage_clear()
490   one_play()
500   game_over()
510 endwhile
520 end
530 /*
540 /*パラメータ初期化
550 /*
560 func initialize()
570   stage=0:count=0:total=0
580   player=maxplayer-1
590   ppos=466
600   locate 42,14: print "         0"
610 endfunc
620 /*
630 /*ゲームオーバー・リプレー処理
640 /*
650 func game_over()
660   locate 11,12 : print "GAME OVER!"
670   locate 4,18 : print "HIT SPACE KEY FOR REPLAY";
680   if inkey$=" " then endflg=1 else endflg=0
690   locate 11,12 : print "          "
700   locate 4,18 : print "
710 endfunc
 720 /*
 730 /*ワンプレー
 740 /*
 750 func one_play()
 760 for pl=1 to player
 770   message()
 780   while move_ball()=1
 790     if by>=base+uwall-bdiam/2*(sgn(sy)+1) and by<=base+uwa
ll+bhight*braw+(sy>0) and ((by-base-uwall) mod bhight)=0 then {
 800     check_block() }
 810     move_paddle()
 820   endwhile
 830   sp_off(1)
 840 next
 850 endfunc
 860 /*
 870 /*プレーヤー
 880 /*
 890 func message()
 900   locate 13,12 : print "READY?"
 910   for iii=1 to 40000 : next
 920   locate 13,12 : print "      "
 930   bx=(rwall+lwall-bdiam)/2 : by=uwall+base+bhight*(braw+2)
 940   sx=0 : sy=spd*4
 950   sp_move(1,bx,by,1)
 960   locate 42,26: print using"##########";player-pl+1
 970 endfunc
 980 /*
 990 /*パドル操作
1000 /*
1010 func move_paddle()
1020   x=read(ch)
1030   if x<lwall+ofs then x=lwall+ofs
1040   if x>rwall-ofs then x=rwall-ofs
1050   if abs(xx-x)>1 then xx=x : px=x-ofs
1060   sp_move(0,px,ppos+bdiam,0)
1070 endfunc
1080 /*
1090 /*ボール運動の判定
1100 /*  （戻り値）1：インプレー
1110 /*           0：ミス
1120 /*
1130 func int move_ball()
1140   bx=bx+sx : by=by+sy
1150   if by>511 then return(0)
1160   if bx<lwall then bx=lwall : sx=-sx
1170   if bx>rwall-bdiam then bx=rwall-bdiam : sx=-sx
1180   if by<lwall then by=lwall : sy=-sy
1190   sp_move(1,bx,by,1)
1200   if by>=ppos and by<=ppos+bhight then check()
1210   return(1)
1220 endfunc
1230 /*
1240 /*パドルにヒットしたかの判定
1250 /*
1260 func check()
1270   if bx>px-bdiam and bx<px+bwidth then {
1280     by=ppos : sy=-sy : sx=int((bx-px-2+sgn(bx-px-4))/4)*sp
d : beep }
1290 endfunc
1300 /*
1310 /*ブロックにヒットしたかの判定
1320 /*
1330 func check_block()
1340   col=(bx+bdiam/2-lwall)¥bwidth+1 : raw=(by-uwall-base)¥bh
ight+1+sgn(sy)
1350   colnxt=col+sgn(sx) : rawnxt=raw-sgn(sy)
1360   if block(col,raw)=1 then { erase_block(col,raw,1) } else
 {
1370     if (bx+bdiam/2+sx-lwall)¥bwidth+1<>col and block(colnx
t,raw)=1 then { erase_block(colnxt,raw,0)
```

```
 1380        if count<>0 and block(colnxt,rawnxt)=1 then {erase_b
lock(colnxt,rawnxt,0) }} }
 1390 endfunc
 1400 /*
 1410 /*ブロック消去
 1420 /*  (引数) 列、行、フラグ
 1430 /*
 1440 func erase_block(col;int,raw;int,flg;int)
 1450   int xcol,yraw,xcol1,yraw1
 1460   block(col,raw)=0
 1470   beep
 1480   xcol=(col-1)*bwidth+lwall : yraw=(raw-1)*bhight+uwall+ba
se
 1490   xcol1=xcol+bwidth-1 : yraw1=yraw+bhight-1
 1500   apage(0)
 1510   fill(xcol,yraw,xcol1,yraw1,0)
 1520   if flg=1 then sy=-sy
 1530   score()
 1540 endfunc
 1550 /*
 1560 /*スプライト(パドル・ボール)定義
 1570 /*
 1580 func spdef()
 1590 sp_color(1,rgb(30,30,30))
 1600 sp_color(2,rgb(10,30,30))
 1610 sp_color(3,rgb(10,30,20))
 1620 dim char paddle(255)={2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2
 1630                      ,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3
 1640                      ,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3
 1650                      ,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3
 1660                      ,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3
 1670                      ,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3
 1680                      ,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3
 1690                      ,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2}
 1700 for iii=128 to 255
 1710    paddle(iii)=0
 1720 next
 1730 sp_def(0,paddle,1)
 1740 dim char ball(255)={0,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0
 1750                   ,0,1,2,2,2,1,0,0,0,0,0,0,0,0,0,0
 1760                   ,1,2,2,2,2,2,1,0,0,0,0,0,0,0,0,0
 1770                   ,1,2,2,2,2,2,1,0,0,0,0,0,0,0,0,0
 1780                   ,1,2,2,2,2,2,1,0,0,0,0,0,0,0,0,0
 1790                   ,1,2,2,2,2,2,1,0,0,0,0,0,0,0,0,0
 1800                   ,0,1,2,2,2,1,0,0,0,0,0,0,0,0,0,0
 1810                   ,0,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0}
 1820 for iii=128 to 255
 1830    ball(iii)=0
 1840 next
 1850 sp_def(1,ball,1)
 1860 endfunc
 1870 /*
 1880 /*壁表示
 1890 /*
 1900 func wall()
 1910   apage(2)
 1920   fill(0,0,10,511,6)
 1930   fill(250,0,260,511,6)
 1940   fill(10,0,250,10,6)
 1950 endfunc
 1960 /*
 1970 /*背景表示
 1980 /*
 1990 func back_ground()
 2000   apage(1)
 2010     line(10,10,39,59,14)
 2020     line(10,59,39,10,14)
 2030     box(10,10,39,59,14)
 2040     fill(15,15,34,54,1)
 2050     get(10,10,39,59,gpat)
 2060   for iii=0 to 7
 2070     for jjj=0 to 9
 2080       cx=iii*30+10 : cy=jjj*50+10
 2090       put(cx,cy,cx+29,cy+49,gpat)
 2100     next
 2110   next
 2120 endfunc
 2130 /*
 2140 /*ブロック表示
 2150 /*
 2160 func draw_block()
 2170   int xcol,yraw,xcol1,yraw1
 2180   apage(0)
 2190   wipe()
 2200   for iii=1 to bcol
 2210     for jjj=1 to braw
 2220       block(iii,jjj)=1
 2230       xcol=(iii-1)*bwidth+lwall : yraw=(jjj-1)*bhight+uwal
l+base
 2240       xcol1=xcol+bwidth-1 : yraw1=yraw+bhight-1
 2250       fill(xcol,yraw,xcol1,yraw1,15)
 2260       fill(xcol+1,yraw+1,xcol1-1,yraw1-1,cl((iii+jjj*3) mo
d 6))
 2270     next
 2280     for jjj= braw+1 to 50
 2290       block(iii,jjj)=0
 2300     next
 2310   next
 2320 endfunc
 2330 /*
 2340 /*スコアー表示
 2350 /*
 2360 func score()
 2370   count=count+1
 2380   total=total+10*stage
 2390   if total>hscore then hscore=total
 2400   locate 42,8 : print using "##########";hscore
 2410   locate 42,14 : print using "##########";total
 2420   if count=braw*bcol then { stage_clear() : message() }
 2430 endfunc
 2440 /*
 2450 /*ステージクリア
 2460 /*
 2470 func stage_clear()
 2480 stage=stage+1 : braw=2*stage
 2490   locate 42,20: print using "##########";stage
 2500 count=0 : player=player+1
 2510 ppos=ppos-bhight*2
 2520 sp_off(1)
 2530 draw_block()
 2540 endfunc
 2550 /*
 2560 /*A/Dコンバータ読み込み
 2570 /*  (引数) チャンネル
 2580 /*  (戻り値) データ値
 2590 /*
 2600 func int read(ch;int)
 2610   int v=0
 2620   ioout(&B1000000)
 2630   start(0,1-ch)
 2640     for iii=0 to 7
 2650       clock(0)
 2660       v=v*2+(ioinp() and 1)
 2670     next
 2680     ioout(&B0)
 2690     return(v)
 2700 endfunc
 2710 /*
 2720 /*A/D変換開始
 2730 /*  (引数) 変換モード、チャンネル
 2740 /*
 2750 func start(sgl;int,sign;int)
 2760   clock(0)
 2770   clock(sgl)
 2780   clock(sign)
 2790 endfunc
 2800 /*
 2810 /*
 2820 /*A/Dコンバータクロック
 2830 /*  (引数) 入力データ
 2840 /*
 2850 func clock(di;int)
 2860   int stat
 2870   stat=&B1000000+di*&B10000000
 2880   ioout(stat+&B0)
 2890   ioout(stat+&B10000)
 2900   ioout(stat+&B0)
 2910 endfunc
```

リスト2 外部関数

```
 1: ************************************************
 2: *                                              *
 3: *     ハードウェア工作入門  I/Oドライバ        *
 4: *          外部関数  ioinp & ioout             *
 5: *                                              *
 6: *      Ver. 2.0    1990.10.20  K. Misawa       *
 7: *                                              *
 8: *        ioinp()      : データ入力             *
 9: *                       (引数)   なし          *
10: *                       (戻り値) データ        *
11: *        ioout(data) : データ出力              *
12: *                       (引数)   データ        *
13: *                       (戻り値) なし          *
14: *                                              *
15: *        adread(ch)  : A/Dコンバータ           *
16: *                         データ読み込み       *
17: *                      (引数) チャンネル       *
18: *                      (戻り値) データ         *
19: *                                              *
20: *      X-BASIC外部関数                         *
21: *         &  BASTOCライブラリ  兼用            *
22: *                                              *
23: ************************************************
24:
25:                 .include        doscall.mac
26:                 .include        fdef.h
27:                 .globl          _ioinp
28:                 .globl          _ioout
29: ************************************************
30:                 .globl          _adread
31: ************************************************
32:                 .text
33:                 .even
34:
35: ************************************************
36: *                                              *
37: *             外部関数ヘッタ部                 *
38: *                                              *
39: ************************************************
40:
41: *
42: *インフォメーションテーブル
43: *
44:                 dc.l            x_init
45:                 dc.l            x_run
46:                 dc.l            x_end
47:                 dc.l            x_sys
48:                 dc.l            x_brk
49:                 dc.l            x_ctrl_d
50:                 dc.l            x_res1
51:                 dc.l            x_res2
52:                 dc.l            ptr_token
53:                 dc.l            ptr_param
54:                 dc.l            ptr_exec
55:                 dc.l            0,0,0,0,0
56:
57: x_init:
58: x_run:
59: x_end:
60: x_sys:
61: x_brk:
62: x_ctrl_d:
63: x_res1:
64: x_res2:
65:
66:                 rts
67: *
68: *関数名テーブル
```

▶来年の1月号にまた付録ディスクをつけるだと？ 受験生の身にもなってみろい！ でも，楽しみ。
原　政徳(17)愛知県

```
 69: *
 70: ptr_token:      dc.b    'ioinp',0
 71:                dc.b    'ioout',0
 72: **************************************************
 73:                dc.b    'adread',0
 74: **************************************************
 75:
 76:                dc.b    0
 77:
 78:                .even
 79:
 80: *
 81: *パラメータテーブル
 82: *
 83: ptr_param:      dc.l    ioinp_par
 84:                dc.l    ioout_par
 85: **************************************************
 86:                dc.l    adread_par
 87: **************************************************
 88:
 89: *
 90: *パラメータIDテーブル
 91: *
 92: ioinp_par:      dc.w    int_ret
 93: ioout_par:      dc.w    int_val
 94:                dc.w    void_ret
 95: **************************************************
 96: adread_par:     dc.w    int_val
 97:                dc.w    int_ret
 98: **************************************************
 99:
100: *
101: *実行アドレステーブル
102: *
103: ptr_exec:       dc.l    ioinp_exec
104:                dc.l    ioout_exec
105: **************************************************
106:                dc.l    adread_exec
107: **************************************************
108:
109:                .even
110:
111: **************************************************
112: *                                                *
113: *               定義関数ルーチン                 *
114: *                                                *
115: **************************************************

116:
117: *
118: *データ入力関数  ioinp()
119: *
120: porta           equ     $e9a001
121:
122: ***実行アドレス
123: ioinp_exec:     bsr     _ioinp
124:                move.l  #0,d0
125:                rts
126:
127: ***メインルーチン
128: _ioinp:
129:
130: ***スーパバイザモードに入る
131:                clr.l   -(sp)
132:                dc.w    _SUPER
133:                addq.l  #4,sp
134:                move.l  d0,spbuf
135:
136: rd_ok:          clr.l   d1
137:
138: ***ジョイスティックポートから読みだし
139:                move.l  #porta,d2
140:                movea.l d2,a3
141:                move.b  (a3),d1
142:
143: ***ユーザーモードに戻る
144:                move.l  spbuf,-(sp)
145:                dc.w    _SUPER
146:                addq.l  #4,sp
147:
148: ***戻り値をバッファに格納
149: rd_ready:       move.l  d1,int_data
150:                lea.l   retdat,a0
151:                move.l  d1,d0
152:
153:                rts
154:
155: *
156: *データ出力関数  ioout(data)
157: *
158: portc           equ     $e9a005
159:
160: ***実行アドレス
161: ioout_exec:
162:
163: ***引数をスタックに積み替え
164:                move.l  12(sp),d1
165:                move.l  d1,-(sp)
166:
167:                bsr     _ioout
168:                addq.l  #4,sp
169:
170:                move.l  #0,d0
171:                rts
172:
173: ***メインルーチン
174: _ioout:
175:
176: ***スーパバイザモードに入る
177:                clr.l   -(sp)
178:                dc.w    _SUPER
179:                addq.l  #4,sp
180:                move.l  d0,spbuf
181:
182: ***ジョイスティックポートへ書き込み
183: wr_ok:          move.l  #portc,d2
184:                movea.l d2,a3
185:                move.l  4(sp),d1
186:                move.b  d1,(a3)
187:
188: ***ユーザーモードに戻る
189:                move.l  spbuf,-(sp)
190:                dc.w    _SUPER
191:                addq.l  #4,sp
192:
193:                rts
194:
195: **************************************************
196: *
197: *A/Dコンバータ読み込み関数  adread(ch)
198: *
199: high            equ     %0
200: low             equ     %10000000
201:
202: ***実行アドレス
203: adread_exec:
204:
205: ***引数をスタックに積み替え
206:                move.l  12(sp),d1
207:                move.l  d1,-(sp)
208:
209:                bsr     _adread
210:                addq.l  #4,sp
211:
212:                move.l  #0,d0
213:                rts
214:
215: ***メインルーチン
216: _adread:        move.l  #%1000000,-(sp)
217:                bsr     _ioout
218:                addq.l  #4,sp
219:
220: ***A/D変換スタート
221:                move.l  #high,-(sp)
222:                bsr     clock
223:                addq.l  #4,sp
224:
225:                move.l  #high,-(sp)
226:                bsr     clock
227:                addq.l  #4,sp
228:
229:                move.l  4(sp),d0
230:                dbra    d0,ch1
231:
232: ***チャンネル0指定
233: ch0:            move.l  #low,-(sp)
234:                bsr     clock
235:                addq.l  #4,sp
236:                bra     read
237:
238: ***チャンネル1指定
239: ch1:            move.l  #high,-(sp)
240:                bsr     clock
241:                addq.l  #4,sp
242:
243: ***シリアルデータ読み込み
244: read:           move.l  #7,d5
245:                clr.l   d6
246:
247: loop:           move.l  #low,-(sp)
248:                bsr     clock
249:                addq.l  #4,sp
250:
251:                bsr     _ioinp
252:                lsl.l   #1,d6
253:                andi.l  #1,d0
254:                add.l   d0,d6
255:
256:                dbra    d5,loop
257:
258: ***戻り値をバッファに格納
259: ready:          move.l  d6,int_data
260:                lea.l   retdat,a0
261:
262:                move.l  #%0,-(sp)
263:                bsr     _ioout
264:                addq.l  #4,sp
265:
266:                move.l  d6,d0
267:
268:                rts
269:
270: ***クロックルーチン(スタックにデータセット)
271: clock:          move.l  4(sp),d3
272:                addi.l  #%1000000,d3
273:                move.l  d3,d4
274:                addi.l  #%10000,d4
275:
276:                move.l  d3,-(sp)
277:                bsr     _ioout
278:                addq.l  #4,sp
279:
280:                move.l  d4,-(sp)
281:                bsr     _ioout
282:                addq.l  #4,sp
283:
284:                move.l  d3,-(sp)
285:                bsr     _ioout
286:                addq.l  #4,sp
287:
288:                rts
289: **************************************************
290:
291: *
292: *スタックバッファ
293: *
294: spbuf           ds.l    1
295:
296: *
297: *戻り値格納バッファ
298: *
299: retdat:         dc.w    0
300:                dc.l    0
301: int_data:       dc.l    0
302:                end
303:
304:
```

▶先日，「満開製作所」からハガキが送られてきました。しかし，家族は「いかがわしい通信販売」としか思っていない。　稲川　裕基(19)岐阜県

アナログジョイスティックの製作

Ishigami Tatsuya 石上 達也

「サイバースティック」を作る。以前ラジコンのプロポからの出力をアナログジョイスティックとして利用する記事がありましたが，今度はスティック部まで自作してみようというもの。はてさて，どうなりますやら。

以前，アナログスティックの製作として，1990年5月号に桒野氏によるものが，発表されました。ラジコンのプロポを入力装置に用い，受信機とインタフェイス回路を接続するという，たいへん独特なものでした。今回，この方法とはまったく別にZ80CPUを用いたコンピュータ制御のアナログスティックを製作しようと思います。

ここでは手持ちのもの，すなわち，PC-8801mkII＋S-OS，ROMライタ，その他工具一式，Z80の知識……を最大限に生かして「できるだけ安く」を目標にしていきます。よってこれらの資材，特にROMライタがない人はかえって手間がかかりますので，桒野氏のCPUを使わない制御回路に積み換えるなど工夫するといいでしょう。

さて，X68000のサイバースティックは「インテリジェントコントローラ」です。この「インテリジェント」というのがクセモノで，4ビットマイコンを使って多彩な機能をまとめているわけです。

我々に馴染みのあるCPU，Z80は8ビット。これも組み込み制御用に多用されているCPUです。おまけに非常に入手しやすく，プログラム開発も容易です。性能にも問題ありません。4ビットでできて8ビットでできないわけはないでしょう。

世のマイコン制御やインテリジェント家電機器の多くは4/8ビットマイコンで制御されています。これらをZ80で代用していくというのは結構面白い試みじゃないでしょうか。

回路について

図1がその回路です。よく見るとRAMがないのがわかると思います。これは制御プログラムを（やっとの思いで）ワークエリアを使わなくても動くようにしたためです(すなわち，LD(nnnn)，AはおろかPUSH&POP，CALL&RETも使っていない)。

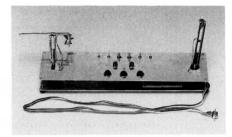
HM6116P-3が200円で買える時代に，なんのために無駄なことをやっているんだ！　と思うかもしれませんが，26ピンのLSIを取り付けるのには，最低52カ所のハンダ付けをしなければなりません。また，RAMを省略することによって，メモリ関係のアドレスデコーダを省略することができるのです。

クロック発振回路は，もっと安価にできないこともないのですが，高価な測定器を持っていないので確実に発振が保証されているクロックオシレータを使用しました。まぁ一応，縁起ものだと思ってください。

そして，もうひとつの縁起ものが，リセット素子のPST518です。リセット素子というのは，電源がふらついたときに，CPUなどに対して，リセット信号を出すものです。この部分も，抵抗とダイオードだけで済ませることも可能なのですが，1個50円でPST518が売っていましたので取り付けておきました。

CPUには，皆さんお馴染みのザイログZ80を用いています。もうこれは，説明する必要がないでしょう。X1やMZシリーズに搭載されているアレです。今回はクロック4MHzで動作させています。これにRAM 48KバイトをつけたらMZ-80Kの約2倍のパフォーマンスを持つことになります。うーん技術の進歩に感謝感謝！

ROMの2764は，アクセスタイムが200ms以下のものを使ってください。ここらへんの回路はえらくシンプルにできました。やはり，制御ソフトのほうでまったくワークエリア（やスタック領域）を使用していないためです。そのため，アドレスデコーダなどが必要なくなり，ほとんどCPUと直結しています。

なぜ，ROMとして2764を使ったのかというと，理由は簡単で私が2764/128のROMライタしか持っていなかったからです。2732とか2716も書き込めるROMライタを持っている方は，そちらを使うとさらに安くあがるでしょう。回路はほとんど変更ありません。回路図のように，ただCPUとROMを直結するだけです。

どことなくサイバーな仕上がり

LS138は，I/O空間用のアドレスデコーダです（さすがにI/O空間用のデコーダは省略できない)。8255AとADC0809の割り振りを行っています。ちなみに，8255Aに対しては，A2のデコードを行っていないので一部影が出てきます。具体的には以下のように割り振っています。

80_H 8255のポートA
81_H 8255のポートB
82_H 8255のポートC
83_H 8255のコントロールワード
84_H 80_Hの影
85_H 81_Hの影
86_H 82_Hの影
87_H 83_Hの影
88_H ADC0809のch0
89_H ADC0809のch1
$8A_H$ ADC0809のch2
$8B_H$ ADC0809のch3
$8C_H$ ADC0809のch4

A/DコンバータにはNS（ナショナルセミコンダクタ）社のADC0809というLSIを使用しました。このLSIは，

1) 5V単一電源で動作可能
2) 8チャンネルの入力端子を持つ
3) TTL，CMOSコンパチ
4) CPUと直結可能

などの特徴を持っています。姉妹品としてA/D変換の精度を上げたADC0808（確か3,000円くらいだったかなぁ？）というLSIもありますが，今回はそんなに精度は必要

ないし，安くアナログスティックを作ろう，とう目標からそれてしまいますのでADC0809で十分です。

下のLS293がCPU用の4MHzのクロックを8分周して500kHzのクロックを作成しています。本当は640kHzあたりが奨励されているんですけど，まぁいいでしょう。

まず，A/Dコンバート開始時にI/Oポート88-8Fのどこかに対してCPUから書き込みが行われます。このとき$\overline{\mathrm{WR}}$と$\overline{\mathrm{SELADC}}$がともにローレベルになりますので，この信号のORをLS04で反転してやり，ADC0809のA/Dコンバート開始要求としてやります。

A/Dコンバートの結果はこれまたI/Oポート88-8F$_H$のどこかに対してCPUが読みにいったとき$\overline{\mathrm{RD}}$と$\overline{\mathrm{SELADC}}$がともにローレベルになるので，これを利用しています。

ここらへんの処理はLS02(NORゲート)などを使うべきなのですが，ICの節約ということで，余ったLS04（NOTゲート）とLS32（ORゲート）を組み合わせて使っています。

図1.1　回路図(その1)

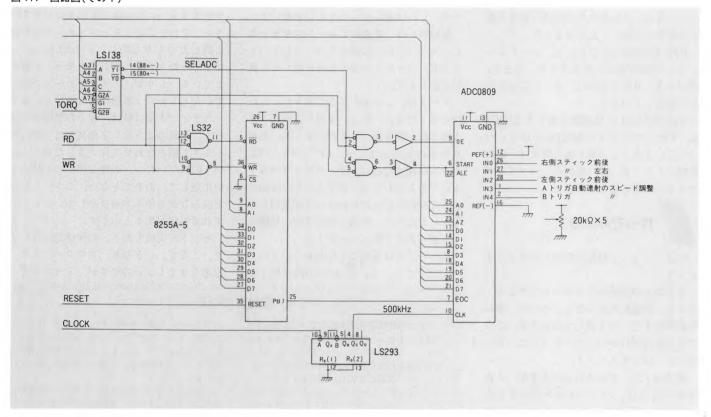



図1.2　回路図(その2)

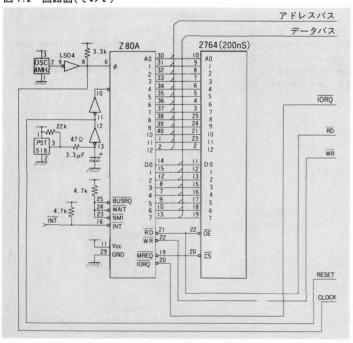
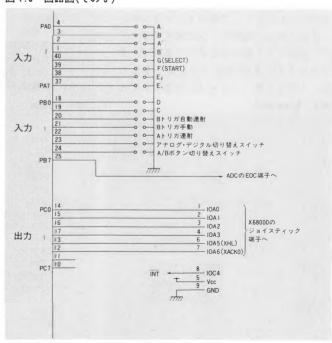
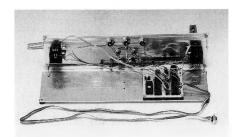



図1.3　回路図(その3)

内部。ボリュームに注目

ちなみに，このあたりの回路は参考文献1を参考にさせていただきました。

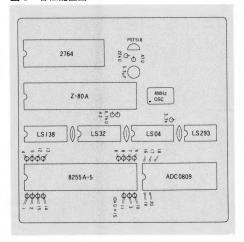
PPI（8255A-5のこと）は，コントロールワード$92を指定してあります。つまり，ポートA，Bは入力用に，ポートCは出力用に使用しています。

X68000からの転送要求命令であるIOC4は，PPIにつながずに直接Z80のINTにつないでいます。この割り込み要求は一定の時間ごとにくるので一種のウォッチドッグタイマの役割も兼ねています。

作ってみよう

では，いよいよ製作に取りかかるとしましょう。

まず図1の回路図をコピーしてきます。そして，配線が終わったところから，順に，色鉛筆などで，塗り潰してゆきます。こうすると配線漏れがなくなりますので，初心者の方にはおすすめです。

回路図では，省略されていますが，それぞれのICには，バイパスコンデンサとして，0.1μF程度のセラミックコンデンサを電源の+5VとGNPピンのあいだに入れておいてください。このバイパスコンデンサというのは，電源ラインに発生するノイズを低減させる働きをします。基板は，サンハヤトのICB504というのを使いましたが，このまま使うとケースに収まりませんので，I行のところで切って使います（基板上には，アルファベットと数字が縦横方向に打ってあり，たいへん便利！）。

ICの配置はどのようでもよいのですが，図2のようにするともっとも配線量が少なくて済むと思います。24ピン以上のLSIには必ずソケットを使ってください。この下にパスコンが収納できます。写真ではクロックオシレータにICソケットを用いてますが，これは別になくてもよいと思います。

配線材には，普通のビニール線を用いるとザルソバみたいに盛り上がってしまいますので，ラッピング用の線材を用いると良いと思います。

ケースは，これまた各自で好きなように作ってください。私は，写真のようにアルミ板2枚を組み合わせて作りました。最初真ん中のネジはなかったのですが，ABCボタンを押すたびに，ケースがペコペコするのであとからつけておきました。今回のような製作にぴったりのケースが見当たらなかったための，苦肉の策ですが，結構安く上がったので気に入ってます。

芯ケーブルは各線ごとに色がついているのがよいでしょう。私はGND線は黒くなきゃ絶対にヤダという人間なので，白い線を外して9芯ケーブルを作りました。コネクタの配線には，十分注意してください。ここがX68000を壊すかもしれない唯一の場所です（ただ，X68000は+5VとGNDをショートさせてもシステムリセットをかけ続けるだけでなんともならないという噂もある……未確認）。

回路図には，表れていませんが，スイッチやボリュームの位置にも気をつけてください。これを間違えると，スティックを右に倒したつもりなのに，F-14は左にいってしまったりとか，デジタルモードで動かしているつもりが，アナログモードで動いていたりとかいう事態に陥ってしまいます。

スティック部は私は，近くの金物屋に行って，写真のようなL字金具をたくさん買ってきて組み合わせて使いましたが，こういうのはだいたい，必要な部品の3倍くらい用意して，ああでもない，こうでもない，と悩みながら積み木遊びをしていると，いずれ妥協案が出るものです。

セットが完成したら，動作確認を行います。ここで，いきなり，アフターバーナーを走らせてもよいのですが，とりあえず，

図2　部品配置図

部品について

今回の製作は電子回路部については特別な部品はありません。問題は機械部です。セット全体でいくらかかるかというのは，だいたいこの部分で差がつきます。私は写真のようにしましたが，ケース部にリプトン紅茶のブリキケース，スロットル部にシャンプーのタンクの柄の部分，スティック部に折れていらなくなったスキーのストックの柄の部分，などというように，身近な廃品を使えば，かなり安くできると思います（私も写真撮影がなかったらそうしたかもしれない）。

まず，写真のようなボリュームを探してきます。今回の製作はこのパーツがないと始まりません（自作してもよいが，かなりハードになりそう）。私は，千石電機で300円で売っているのを買った直後，秋月電子で参考回路図付きで200円で売っているのを見つけました。一瞬，まったく同じものかと思いましたが，前者のほうが，やや丈夫なようでした。ここは，かなりの力がかかるところですので，なるべく丈夫なものがよいでしょう。

お金に糸目をつけなければ（このセットの製作意図と矛盾しますが），電動車椅子取り付け用のポテンショメーターだとか，産業ロボット操作用のスティックなんかあればベターです。もし，中古のパワーショベルとか，米空軍払い下げのF-15のコックピットなんかあればベストです。もし，これらの部品が手に入れば，本家本元のサイバースティックよりも丈夫なものができます（入手に成功した方はぜひ編集部までご一報ください）。

表1：部品表

品名	数量・値段
抵抗	
47Ω	
3.3k	
4.7k	×2
10k	×15
22k	
可変抵抗	
20k（B型）	×　2（回転型）
20k（B型）	×　2（写真のようなもの）
コンデンサ	
電解　3.3μF	
その他パスコンとして0.1μFくらいのセラミックコンデンサを8個	
IC	値段
74LS04	30
74LS32	40
74LS138	55
74LS293	65
Z80A	210
8255A	180
2764-20	800
ADC0809	900
リセット素子　PST518	50
クロックオシレータ　4.0000MHz	800
基板	
サンハヤト　ICB504	210
スイッチ類	
マイクロスイッチ	×5
押しボタンスイッチ	×5
トグルスイッチ（2P）	×4
トグルスイッチ（3P）	×1
アルミ板　2t×400×100	×2
ゴム足	×5

1990年5月号の桒野氏のプログラムを使って確認しましょう。

このとき，アナログ/デジタル切り替えスイッチはアナログ側にしておかないとテストが行えませんので注意してください（私だけが，こんな点にうっかりするのかなぁ？）。

ここで，A/B切り替えスイッチや，連射スイッチの動作を確かめ，ボタンの付け間違いがないかを確認しておいてください。

アナログモードで動けば，ROMの書き間違いでもしない限りデジタルモードは動作するのですが，一応確認するというか，完成の喜びに浸るという意味でグラディウスでも遊んでみましょう。

ここまできたら，あとはもう，アフターバーナーでもサンダーブレードでも好きなだけ遊んでみましょう。

ソフトについて

それではいよいよ制御用ソフトの説明に入りたいと思います。

最初，このセットはPC－8001mkIIを使って組んでありました。そして，制御プログラムの開発はここから行いました。PC上では，Z80は原則的にモード2で動作する，とか，DMAが一定周期でかかってくるので動作速度が違う，だとかいろいろありましたけれど，結局，組み込みコンピュータの開発としてはこのスタイルが一番なのではないかと思います。そりゃ，ICEがあれば，BESTだけど，あれって1台安くても20万円ぐらいするんですよ。

先ほどから何度も述べましたように，今回のセットには経費削減のため，じゃなくてコンパクト化のためRAMを積んでいません。すべての情報はZ80がレジスタに覚えています。その内訳はだいたい以下のとおりです。

B：Aボタン連射時のカウンタ

C：Bボタン連射時のカウンタ

D上位：ABボタンのデータ（1回目に送る分）

D下位：ABボタンのデータ（7回目に送る分）

E：右側スティックの前後角度

H：右側スティックの左右角度

L：左側スティックの前後角度

リストを見ていただければわかると思いますが，大変似かよったところが多々見うけられます。これはRAMを持たないため，サブルーチンコールをできないばかりでなく，ワークエリアを一切持てないので，「PC (Programming Counter) がいまどこにいるか」，ということ自体にある種の情報を持たせているのです。具体的には，PCがラベル「anaR……」というところにいれば，もうそれでアナログモードでA/Bボタン判定スイッチが有効(Reverse Mode)ということを意味します。

アナログモードにおいて，X68000から転送要求があると，Z80はそのとき行っている仕事を止めて無条件に0038_H番地へ飛びます。するとここには，

```
JP      SEND
```

という命令が書かれていますので，データ転送ルーチンSEND：へ飛びます。本来なら，SEND：での仕事が終わると，CPUは元の仕事に戻るべきですが，今回のセットではRAMを積んでいないので戻るべきアドレス（元の仕事のありか）を記憶しておけません。

そこで今回は強引にSEND：での仕事が終わるとCPUは必ずプログラムの一番初めに戻ることにしました。つまりX68000からの転送要求命令はZ80のリセット命令の役割も兼ねているのです。

今回のセットではデータの転送速度は固定です。わざわざ，組み込み型コンピュータを入れておいてそれは手抜きじゃないの，と思うかもしれませんが，そうです。手抜きです。とりあえずのところ，X68000やPC－9801がアナログスティックの動作に追いつかないなんてことはないと思いますので安心してください。

さらにばらしてしまうと，転送スピードもあまり厳密ではありません。これも手抜きです。結構ラフにデータを垂れ流しても，ちゃんとX68000が拾ってくれるような転送方式ですので心配はないと思います。

0000_H番地付近と0038_H番地付近に，

```
JP   0FFFFH
```

というのが並んでいますが，これはROMを何回か有効に利用しようとするものです。ROMの書き込みとは，ROMの初期値である$0FF_H$に対して，どこぞのビットを0にしていく，つまり，書き込みたい値と論理和を取っていくということです。

いったんバグの抜け切らないプログラムをROMに書き込んでしまっても，次のプログラムでは，その領域を，

```
00  (=NOP)
```

で埋めておけば，ROMを消去することなしに使用することができます（本当はROMっぽく動くRAMで作ったROMエミュレータというのを持っていたんだけど，壊れちゃったんだようぅ……）。

2回目以降はリセットされてくる0000_H番地と割り込みがかかってくるとコールされる0038_H番地は00_Hで埋めて，代わりに，その次の0003_H番地と$003B_H$番地のジャンプ命令を有効にすればよいのです。

後日談

「サイバースティックの23,800円は許せない。たかだかジョイスティックだろうが!」というのが，今回の製作の動機でした。しかし，実際に作ってみて「いやぁ～23,800円というのは，なかなか良心的じゃあないかい？」というのが感想です。

電子回路の部分は，まぁまぁ満足のいく値段に抑えることができたのですが，機械

A/Dコンバータについて

アナログな情報をコンピュータで扱えるようなデジタル情報に変換するインタフェイスのことをA/Dコンバータ（Analog Digital Converter）と呼びます。

今回の製作ではジョイスティクの傾き具合をボリュームを用いてアナログ量で表し，それをZ80CPUで処理しています。

A/Dコンバータには大きく分けて積分型のものと逐次比較型のものと，2通りあります。

積分型とは，まず一定時間，測りたい電圧を積分用のコンデンサに加えます。次にそれとは逆の方向に基準電圧を加えていきます。そして，積分用のコンデンサの電荷がなくなるまでの時間をカウントしてやります。この方法は，精度が高く取れるのですが，変換に時間がかかったり，変換用のLSIが高かったりするので今回は見送りました。

もう一方の逐次比較型とは，まず自分で比較電圧を作成し，その電圧と入力電圧の大きさをアナログコンパレータ（比較器）を用いてどちらが大きいかを判断し，状況に応じて，また比較電圧を作成し比較をしていくというものです。具体的にいうと，まず，基準電圧の1/2と入力電圧を比較します。もし，入力電圧が大きければ最上位ビット（MSB）は1で，小さければ0です。次にその電圧の差ΔV1を基準電圧の1/4と比較します。もし，ΔV1のほうが大きければ，次のビットが1になり，小さければ0になります。そして，その差ΔV2と基準電圧の1/8を比較して……，というように入力電圧の大きさを決定します。

今回使用したADC0809は8ビットの分解能（2の8乗＝256段階）を持っていますので，この比較を8回行います。

あ，そうそう，可変抵抗器全般を指してボリュームというのは日本独自の英語だそうで，本当の正しい(?)英語では，Variable Resisterとか，Potentio Meterとかいうのだそうです。VolumeだったらVLですけど，回路図なんかではボリュームのことをVRと書きますよね。

部分が思ったより高くなってしまいました。それで高いお金をかけた分，よいものが出来上がったかというと，そうでもないようです。私のセットではアフターバーナーを1周すると，右手の親指が痛くなってしまいます。これも各自の工夫次第。サイバースティックに飽き足らない人，金はないがハードはまかしとけという人は参考にしてください（参考になるかな……)。

参考文献：

1）作りながら学ぶマイコン設計トレーニング（CQ出版） 神崎 康弘

2）Oh!X 1990年5月号 ラジコンスティックの製作 桒野 雅彦

ウォッチドッグタイマ

コンピュータでなにかものを制御する場合，コンピュータの暴走に対しては，かなり気をつけなければなりません。パソコンなどの場合でしたらだいたい暴走したなぁ，とかわかるのですが組み込みコンピュータの場合だとそれを外部から知る術がありません。

1回暴走したら，それこそ電源を切ってやるまで人知れず黙々と暴走を続けるのです。それなら，いっそのこと電源を切らないまでも，一定時間ごとにシステムにリセットをかけてやれば，たとえ暴走してもその暴走している時間は限られてきます。そのリセットをかけてやるのが，ウォッチドッグタイマ（Watch Dog：番犬）回路なのです。

今回のアナログスティックの場合，特別なウォッチドッグタイマ回路は搭載していませんがX68000からの転送要求命令をこれに代用しています。どういうことかというと，普通は割り込み処理ルーチンでの仕事が終わったら，いままでやっていた仕事の続きをするものなのですが，今回のアナログスティックはRAMを積んでいません。すなわち，いままでの仕事に戻ろうにも戻るべき仕事を記憶していられないのです。そこで割り込み処理ルーチンから抜ける際には常にシステムの最初の状態，すなわち電源投入時と同じリセットがかかるようにしてやるのです。

転んでも，決してタダでは起きない設計だと自分では思っています。

リスト

```
0000                1  ;********************************
0000                2  ;
0000                3  ; Cyber Stick Control Program
0000                4  ;        Programmed By T.Ishigami
0000                5  ;                 '90 Jul 31th
0000                6  ;
0000                7  ;********************************
0000                8
0080 P              9  PORTA   EQU     80H
0081 P             10  PORTB   EQU     81H
0082 P             11  PORTC   EQU     82H
0083 P             12  CW      EQU     83H
0000               13
0088 P             14  CH0     EQU     88H
0089 P             15  CH1     EQU     89H
008A P             16  CH2     EQU     8AH
008B P             17  CH3     EQU     8BH
008C P             18  CH4     EQU     8CH
0000               19
0000               20          OFFSET  8000H
0000               21          ORG     0000H
0000               22
0000               23          DS      16H             ;Masking Pater
n
0016               24
0016 C3 44 00      25          JP      START
0019               26
0019 C3 FF FF      27          JP      0FFFFH
001C C3 FF FF      28          JP      0FFFFH
001F C3 FF FF      29          JP      0FFFFH          ;Reserved For
Future
0022               30
0038               31          ORG     0038H
0038               32
0038 C3 E0 02      33          JP      SEND
003B               34
003B C3 FF FF      35          JP      0FFFFH
003E C3 FF FF      36          JP      0FFFFH
0041 C3 FF FF      37          JP      0FFFFH          ;Reserved For
Future
0044               38
0044               39
0044               40
0044 ED 56         41  START:  IM      1               ;Interrupt Mod
e 1
0046               42
0046 3E 92         43          LD      A,92H
0048 D3 83         44          OUT     (CW),A
004A               45
004A               46  digital:
004A F3            47          DI
004B               48
004B DB 81         49          IN      A,(PORTB)
004D CB 6F         50          BIT     5,A
004F CA 18 01      51          JP      Z,analog
0052               52
0052               53  ; Digital Mode
0052               54
0052 D3 8B         55  digA:   OUT     (CH3),A         ;CH3 ｦ ｱﾗｶｼﾞﾒ
CONVERT ｼﾃｵｸ
0054               56
0054 DB 80         57          IN      A,(PORTA)
0056 CB 47         58          BIT     0,A
0058 28 0A         59          JR      Z,digA1         ;if A = ON
005A CB 57         60          BIT     2,A
005C 28 06         61          JR      Z,digA1         ;if A'= ON
005E 26 00         62          LD      H,0
0060 06 FF         63          LD      B,0FFH          ;ﾚﾝｼｬ ｼﾊｼﾞﾒﾊ ｶ
ﾅﾗｽﾞ A = ON ﾆ ﾅﾙﾖｳﾆ
0062               64
0062 18 21         65          JR      digB
0064               66
0064 DB 81         67  digA1:  IN      A,(PORTB)
0066 CB 67         68          BIT     4,A
0068 28 19         69          JR      Z,digA4
006A               70
006A DB 81         71  digA3:  IN      A,(PORTB)       ;ｼﾞﾄﾞｳ ﾚﾝｼｬ
006C CB 7F         72          BIT     7,A
006E 28 FA         73          JR      Z,digA3         ;Wait   EOC =
1
0070               74
0070 DB 8B         75          IN      A,(CH3)
0072 CB 3F         76          SRL     A
0074 CB 3F         77          SRL     A
0076 CB 3F         78          SRL     A
0078 3C            79          INC     A               ; A = (CH3) /
8 + 1
0079               80
0079 80            81          ADD     A,B
007A 47            82          LD      B,A
007B 30 08         83          JR      NC,digB
007D               84
007D 7C            85          LD      A,H
007E EE 01         86          XOR     1
0080 67            87          LD      H,A
0081 18 02         88          JR      digB
0083               89
0083 26 01         90  digA4:  LD      H,1             ;Normal
0085               91
0085 D3 8C         92  digB:   OUT     (CH4),A         ;CH4 ｦ ｱﾗｶｼﾞﾒ
CONVERT ｼﾃｵｸ
0087               93
0087 DB 81         94          IN      A,(PORTB)
0089 CB 57         95          BIT     2,A
008B 28 16         96          JR      Z,digB2         ;button B Set
AUTO MODE
008D               97
008D DB 80         98          IN      A,(PORTA)
008F CB 4F         99          BIT     1,A
0091 28 0A        100          JR      Z,digB1         ;if B = ON
0093 CB 5F        101          BIT     3,A
0095 28 06        102          JR      Z,digB1         ;if B'= ON
0097 2E 00        103          LD      L,0
0099 0E FF        104          LD      C,0FFH          ;ﾚﾝｼｬ ｼﾊｼﾞﾒﾊ ﾂ
ｷﾆ B = ON
ﾆ ﾅﾙﾖｳﾆ
009B 18 23        105          JR      digY
009D              106
009D DB 81        107  digB1:  IN      A,(PORTB)
009F CB 5F        108          BIT     3,A
00A1 28 1B        109          JR      Z,digB3
00A3              110
00A3 DB 81        111  digB2:  IN      A,(PORTB)       ;ｼﾞﾄﾞｳ ﾚﾝｼｬ
00A5 CB 7F        112          BIT     7,A
00A7 28 FA        113          JR      Z,digB2         ;Wait   EOC =
1
00A9              114
00A9 DB 8C        115          IN      A,(CH4)
00AB CB 3F        116          SRL     A
00AD CB 3F        117          SRL     A
00AF CB 3F        118          SRL     A
00B1 CB 3F        119          SRL     A
00B3 3C           120          INC     A               ;A = (CH4) / 8
 +1
00B4              121
00B4 81           122          ADD     A,C
00B5 4F           123          LD      C,A
00B6 30 08        124          JR      NC,digY
00B8              125
00B8 7D           126          LD      A,L
00B9 EE 01        127          XOR     1
00BB 6F           128          LD      L,A
00BC 18 02        129          JR      digY
00BE              130
00BE 2E 01        131  digB3:  LD      L,1
00C0              132
00C0              133
00C0 D3 88        134  digY:   OUT     (CH0),A
00C2              135
00C2 DB 81        136  digY1:  IN      A,(PORTB)
00C4 CB 7F        137          BIT     7,A
00C6 28 FA        138          JR      Z,digY1                 ;Wait
EOC = 1
00C8              139
00C8 DB 88        140          IN      A,(CH0)
00CA              141
00CA 16 FE        142          LD      D,0FEH
00CC FE 64        143          CP      100
00CE 38 08        144          JR      C,digX
00D0 16 FD        145          LD      D,0FDH
00D2 FE 8C        146          CP      140
00D4 30 02        147          JR      NC,digX
00D6 16 FF        148          LD      D,0FFH
00D8              149
00D8 D3 89        150  digX:   OUT     (CH1),A
00DA              151
00DA DB 81        152  digX1:  IN      A,(PORTB)
00DC CB 7F        153          BIT     7,A
00DE 28 FA        154          JR      Z,digX1
00E0              155
00E0 DB 89        156          IN      A,(CH1)
00E2              157
00E2 FE 64        158          CP      100
00E4 38 06        159          JR      C,digX2
00E6 FE 8C        160          CP      140
00E8 30 06        161          JR      NC,digX3
00EA 18 06        162          JR      DOUT
00EC              163
00EC CB 92        164  digX2:  RES     2,D
00EE 18 02        165          JR      DOUT
00F0 CB 9A        166  digX3:  RES     3,D
00F2              167
00F2              168
00F2 DB 81        169  DOUT:   IN      A,(PORTB)
00F4 CB 77        170          BIT     6,A
00F6 20 0E        171          JR      NZ,DOUT3        ;A / B ﾊﾝﾃﾝ ?
00F8              172
00F8 7C           173          LD      A,H             ;Normal
00F9 A7           174          AND     A
00FA 28 02        175          JR      Z,DOUT1
00FC CB A2        176          RES     4,D
00FE 7D           177  DOUT1:  LD      A,L
00FF A7           178          AND     A
0100 28 02        179          JR      Z,DOUT2
0102 CB AA        180          RES     5,D
0104 18 0C        181  DOUT2:  JR      DOUT5
0106              182
0106 7C           183  DOUT3:  LD      A,H             ;Reverse
0107 A7           184          AND     A
0108 28 02        185          JR      Z,DOUT4
010A CB AA        186          RES     5,D
010C 7D           187  DOUT4:  LD      A,L
010D A7           188          AND     A
010E 28 02        189          JR      Z,DOUT5
0110 CB A2        190          RES     4,D
0112              191
0112 7A           192  DOUT5:  LD      A,D
0113 D3 82        193          OUT     (PORTC),A
0115              194
0115 C3 4A 00     195          JP      digital
0118              196
0118              197
0118              198  ;
0118              199  ; Analog Mode
0118              200  ;
0118              201  analog:
0118 FB           202          EI
0119              203
0119 DB 81        204          IN      A,(PORTB)
011B              205
011B CB 6F        206          BIT     5,A
011D C2 4A 00     207          JP      NZ,digital      ;digital Mode
0120              208
0120 CB 77        209          BIT     6,A
0122 C2 EC 01     210          JP      NZ,anaRA        ;A / B Reverse
d
0125              211
0125              212  ;
0125              213  ; Analog & Normal Mode
0125              214  ;
0125 D3 8B        215  anaNA:  OUT     (CH3),A         ;CH3 ｦ ｱﾗｶｼﾞﾒ
CONVERT ｼﾃｵｸ
0127              216
0127 DB 80        217          IN      A,(PORTA)
0129 CB 47        218          BIT     0,A
012B 28 0D        219          JR      Z,anaNA1        ;if A = ON
012D CB 57        220          BIT     2,A
012F 28 2F        221          JR      Z,anaNA5        ;if A'= ON
0131              222
0131 06 FF        223          LD      B,-1            ;ﾚﾝｼｬ ｼﾏｼﾞﾒﾊ ｶ
ﾅﾗｽﾞ A = ON ﾆ ﾅﾙﾖｳﾆ
0133 3E 8A        224          LD      A,8AH
0135 B2           225          OR      D
0136 57           226          LD      D,A             ;Mask 1*** 1*1
*
0137 C3 83 01     227          JP      anaNB
013A              228
013A              229
013A DB 81        230  anaNA1: IN      A,(PORTB)       ;A = ON
013C CB 67        231          BIT     4,A
013E 28 19        232          JR      Z,anaNA3
0140              233
0140 DB 81        234  anaNA2: IN      A,(PORTB)       ;ｼﾞﾄﾞｳ ﾚﾝｼｬ (
A )
0142 CB 7F        235          BIT     7,A
0144 28 FA        236          JR      Z,anaNA2                ;Wait
  EOC = 1
0146              237
0146 DB 8B        238          IN      A,(CH3)
0148 CB 3F        239          SRL     A
014A CB 3F        240          SRL     A
014C CB 3F        241          SRL     A
014E 3C           242          INC     A               ; A = (CH3) /
8 + 1
014F              243
014F 80           244          ADD     A,B
0150 47           245          LD      B,A
0151 30 0D        246          JR      NC,anaNA5
0153              247
0153 7A           248          LD      A,D
0154 EE 88        249          XOR     88H             ;(A+A'),(A) ﾆ
ﾂｲﾃ
0156 57           250          LD      D,A
0157 18 07        251          JR      anaNA5
0159              252
0159 3E 77        253  anaNA3: LD      A,77H           ;Normal
015B A2           254          AND     D               ;Mask 0***0***
015C 57           255          LD      D,A
015D C3 83 01     256          JP      anaNB
0160              257
0160 DB 81        258  anaNA5: IN      A,(PORTB)       ;A' = ON
0162 CB 67        259          BIT     4,A
0164 28 19        260          JR      Z,anaNA7
0166              261
0166 DB 81        262  anaNA6: IN      A,(PORTB)       ;ｼﾞﾄﾞｳ ﾚﾝｼｬ (
A )
0168 CB 7F        263          BIT     7,A
016A 28 FA        264          JR      Z,anaNA6                ;Wait
  EOC = 1
016C              265
016C DB 8B        266          IN      A,(CH3)
016E CB 3F        267          SRL     A
0170 CB 3F        268          SRL     A
0172 CB 3F        269          SRL     A
0174 3C           270          INC     A               ; A = (CH3) /
8 + 1
0175              271
0175 80           272          ADD     A,B
0176 47           273          LD      B,A
0177 30 0A        274          JR      NC,anaNB
0179              275
0179 7A           276          LD      A,D
017A EE 82        277          XOR     82H
017C 57           278          LD      D,A
017D 18 04        279          JR      anaNB
017F              280
017F 3E 7D        281  anaNA7: LD      A,7DH           ;Normal
0181 A2           282          AND     D               ;Mask 0*** **0
*
0182 57           283          LD      D,A
0183              284
0183              285
0183 D3 8C        286  anaNB:  OUT     (CH4),A         ;CH4 ｦ ｱﾗｶｼﾞﾒ
CONVERT ｼﾃｵｸ
0185              287
0185 DB 80        288          IN      A,(PORTA)
0187 CB 4F        289          BIT     1,A
0189 28 13        290          JR      Z,anaNB1        ;if B = ON
018B CB 5F        291          BIT     3,A
018D 28 35        292          JR      Z,anaNB5        ;if B'= ON
018F DB 81        293          IN      A,(PORTB)
0191 CB 57        294          BIT     2,A
0193 28 0F        295          JR      Z,anaNB2        ;if B AUTO = O
N
0195              296
0195 0E FF        297          LD      C,-1            ;ﾚﾝｼｬ ｼﾏｼﾞﾒﾊ ｶ
ﾅﾗｽﾞ A = ON ﾆ ﾅﾙﾖｳﾆ
```

```
0197 3E 45          298             LD      A,45H
0199 B2             299             OR      D
019A 57             300             LD      D,A             ;Mask *1** *1*
1
019B C3 AE 02       301             JP      anaC
019E                302
019E                303
019E DB 81          304     anaNB1: IN      A,(PORTB)       ;B = ON
01A0 CB 5F          305             BIT     3,A
01A2 28 19          306             JR      Z,anaNB3
01A4                307
01A4 DB 81          308     anaNB2: IN      A,(PORTB)       ;ｼﾞﾄﾞｳ ﾚﾝｼｬ (
B )
01A6 CB 7F          309             BIT     7,A
01A8 28 FA          310             JR      Z,anaNB2        ;Wait   EOC =
1
01AA                311
01AA DB 8C          312             IN      A,(CH4)
01AC CB 3F          313             SRL     A
01AE CB 3F          314             SRL     A
01B0 CB 3F          315             SRL     A
01B2 3C             316             INC     A               ; A = (CH4) /
8 + 1
01B3                317
01B3 81             318             ADD     A,C
01B4 4F             319             LD      C,A
01B5 30 0D          320             JR      NC,anaNB5
01B7                321
01B7 7A             322             LD      A,D
01B8 EE 44          323             XOR     44H             ;(B+B'),(B) ｦ
ﾂｲﾃ
01BA 57             324             LD      D,A
01BB 18 07          325             JR      anaNB5
01BD                326
01BD 3E BB          327     anaNB3: LD      A,0BBH          ;Normal
01BF A2             328             AND     D               ;Mask *0** *0*
*
01C0 57             329             LD      D,A
01C1 C3 AE 02       330             JP      anaC
01C4                331
01C4 DB 81          332     anaNB5: IN      A,(PORTB)       ;B' = ON
01C6 CB 5F          333             BIT     3,A
01C8 28 1B          334             JR      Z,anaNB7
01CA                335
01CA DB 81          336     anaNB6: IN      A,(PORTB)       ;ｼﾞﾄﾞｳ ﾚﾝｼｬ (
B )
01CC CB 7F          337             BIT     7,A
01CE 28 FA          338             JR      Z,anaNB6        ;Wait   EOC =
1
01D0                339
01D0 DB 8C          340             IN      A,(CH4)
01D2 CB 3F          341             SRL     A
01D4 CB 3F          342             SRL     A
01D6 CB 3F          343             SRL     A
01D8 3C             344             INC     A               ; A = (CH4) /
8 + 1
01D9                345
01D9 81             346             ADD     A,C
01DA 4F             347             LD      C,A
01DB D2 AE 02       348             JP      NC,anaC
01DE                349
01DE 7A             350             LD      A,D
01DF EE 41          351             XOR     41H
01E1 57             352             LD      D,A
01E2 C3 AE 02       353             JP      anaC
01E5                354
01E5 3E BE          355     anaNB7: LD      A,0BEH          ;Normal
01E7 A2             356             AND     D               ;Mask *0** ***
0
01E8 57             357             LD      D,A
01E9                358
01E9 C3 AE 02       359             JP      anaC
01EC                360
01EC                361     ;
01EC                362     ; Analog & A / B Reversed Mode
01EC                363     ;
01EC D3 8B          364     anaRA:  OUT     (CH3),A         ;CH3 ｦ ｱﾗｶｼﾞﾒ
CONVERT ｼﾃｵｸ
01EE                365
01EE DB 80          366             IN      A,(PORTA)
01F0 CB 47          367             BIT     0,A
01F2 28 0D          368             JR      Z,anaRA1                ;if A
= ON
01F4 CB 57          369             BIT     2,A
01F6 28 2F          370             JR      Z,anaRA5                ;if A'
= ON
01F8                371
01F8 06 FF          372             LD      B,-1            ;ﾚﾝｼｬ ｼﾏｼﾞﾒﾊ ｶ
ﾅﾗｽﾞ A = ON ﾆ ﾅﾙﾖｳﾆ
01FA 3E 45          373             LD      A,45H
01FC B2             374             OR      D
01FD 57             375             LD      D,A             ;Mask *1** *1*
1
01FE C3 4A 02       376             JP      anaRB
0201                377
0201                378
0201 DB 81          379     anaRA1: IN      A,(PORTB)       ;A = ON
0203 CB 67          380             BIT     4,A
0205 28 19          381             JR      Z,anaRA3
0207                382
0207 DB 81          383     anaRA2: IN      A,(PORTB)       ;ｼﾞﾄﾞｳ ﾚﾝｼｬ (
A )
0209 CB 7F          384             BIT     7,A
020B 28 FA          385             JR      Z,anaRA2                ;Wait
 EOC = 1
020D                386
020D DB 8B          387             IN      A,(CH3)
020F CB 3F          388             SRL     A
0211 CB 3F          389             SRL     A
0213 CB 3F          390             SRL     A
0215 3C             391             INC     A               ; A = (CH3) /
8 + 1
0216                392
0216 80             393             ADD     A,B
0217 47             394             LD      B,A
0218 30 0D          395             JR      NC,anaRA5
021A                396
021A 7A             397             LD      A,D
021B EE 44          398             XOR     44H             ;(A+A'),(A) ｦ
ﾂｲﾃ
021D 57             399             LD      D,A
021E 18 07          400             JR      anaRA5
0220                401
0220 3E BB          402     anaRA3: LD      A,0BBH          ;Normal
0222 A2             403             AND     D               ;Mask *0** *0*
*
0223 57             404             LD      D,A
0224 C3 4A 02       405             JP      anaRB
0227                406
0227 DB 81          407     anaRA5: IN      A,(PORTB)       ;A' = ON
0229 CB 67          408             BIT     4,A
022B 28 19          409             JR      Z,anaRA7
022D                410
022D DB 81          411     anaRA6: IN      A,(PORTB)       ;ｼﾞﾄﾞｳ ﾚﾝｼｬ (
A )
022F CB 7F          412             BIT     7,A
0231 28 FA          413             JR      Z,anaRA6                ;Wait
 EOC = 1
0233                414
0233 DB 8B          415             IN      A,(CH3)
0235 CB 3F          416             SRL     A
0237 CB 3F          417             SRL     A
0239 CB 3F          418             SRL     A
023B 3C             419             INC     A               ; A = (CH3) /
8 + 1
023C                420
023C 80             421             ADD     A,B
023D 47             422             LD      B,A
023E 30 0A          423             JR      NC,anaRB
0240                424
0240 7A             425             LD      A,D
0241 EE 41          426             XOR     41H
0243 57             427             LD      D,A
0244 18 04          428             JR      anaRB
0246                429
0246 3E BE          430     anaRA7: LD      A,0BEH          ;Normal
0248 A2             431             AND     D               ;Mask *0** ***
0
0249 57             432             LD      D,A
024A                433
024A                434
024A D3 8C          435     anaRB:  OUT     (CH4),A         ;CH4 ｦ ｱﾗｶｼﾞﾒ
CONVERT ｼﾃｵｸ
024C                436
024C DB 80          437             IN      A,(PORTA)
024E CB 4F          438             BIT     1,A
0250 28 13          439             JR      Z,anaRB1                ;if B
= ON
0252 CB 5F          440             BIT     3,A
0254 28 35          441             JR      Z,anaRB5                ;if B'
= ON
0256 DB 81          442             IN      A,(PORTB)
0258 CB 57          443             BIT     2,A
025A 28 0F          444             JR      Z,anaRB2                ;if B
AUTO = ON
025C                445
025C 0E FF          446             LD      C,-1            ;ﾚﾝｼｬ ｼﾏｼﾞﾒﾊ ｶ
ﾅﾗｽﾞ A = ON ﾆ ﾅﾙﾖｳﾆ
025E 3E 8A          447             LD      A,8AH
0260 B2             448             OR      D
0261 57             449             LD      D,A             ;Mask 1*** 1*1
*
0262 C3 AE 02       450             JP      anaC
0265                451
0265                452
0265 DB 81          453     anaRB1: IN      A,(PORTB)       ;B = ON
0267 CB 5F          454             BIT     3,A
0269 28 19          455             JR      Z,anaRB3
026B                456
026B DB 81          457     anaRB2: IN      A,(PORTB)       ;ｼﾞﾄﾞｳ ﾚﾝｼｬ (
B )
026D CB 7F          458             BIT     7,A
026F 28 FA          459             JR      Z,anaRB2                ;Wait
 EOC = 1
0271                460
0271 DB 8B          461             IN      A,(CH3)
0273 CB 3F          462             SRL     A
0275 CB 3F          463             SRL     A
0277 CB 3F          464             SRL     A
0279 3C             465             INC     A               ; A = (CH3) /
8 + 1
027A                466
027A 81             467             ADD     A,C
027B 4F             468             LD      C,A
027C 30 0D          469             JR      NC,anaRB5
027E                470
027E 7A             471             LD      A,D
027F EE 88          472             XOR     88H             ;(B+B'),(B) ｦ
ﾂｲﾃ
0281 57             473             LD      D,A
0282 18 07          474             JR      anaRB5
0284                475
0284 3E 77          476     anaRB3: LD      A,77H           ;Normal
0286 A2             477             AND     D               ;Mask 0*** 0**
*
0287 57             478             LD      D,A
0288 C3 AE 02       479             JP      anaC
028B                480
028B DB 81          481     anaRB5: IN      A,(PORTB)       ;B' = ON
028D CB 5F          482             BIT     3,A
028F 28 19          483             JR      Z,anaRB7
0291                484
0291 DB 81          485     anaRB6: IN      A,(PORTB)       ;ｼﾞﾄﾞｳ ﾚﾝｼｬ (
B )
0293 CB 7F          486             BIT     7,A
0295 28 FA          487             JR      Z,anaRB6        ;Wait   EOC =
1
0297                488
0297 DB 8C          489             IN      A,(CH4)
0299 CB 3F          490             SRL     A
029B CB 3F          491             SRL     A
029D CB 3F          492             SRL     A
029F 3C             493             INC     A               ; A = (CH4) /
8 + 1
02A0                494
02A0 81             495             ADD     A,C
02A1 4F             496             LD      C,A
02A2 30 0A          497             JR      NC,anaC
02A4                498
02A4 7A             499             LD      A,D
02A5 EE 82          500             XOR     82H
02A7 57             501             LD      D,A
02A8 18 04          502             JR      anaC
02AA                503
02AA 3E 7D          504     anaRB7: LD      A,7DH           ;Normal
02AC A2             505             AND     D               ;Mask 0*** **0
*
02AD 57             506             LD      D,A
02AE                507
02AE                508
02AE CB E2          509     anaC:   SET     4,D
02B0 CB EA          510             SET     5,D
02B2                511
02B2 DB 81          512             IN      A,(PORTB)
02B4 87             513             ADD     A,A
02B5 87             514             ADD     A,A
02B6 87             515             ADD     A,A
02B7 87             516             ADD     A,A             ; A = A * 16
02B8                517
02B8 F6 CF          518             OR      0CFH
02BA A2             519             AND     D
02BB 57             520             LD      D,A
02BC                521
02BC D3 88          522             OUT     (CH0),A
02BE DB 81          523     anaCH0: IN      A,(PORTB)
02C0 CB 7F          524             BIT     7,A
02C2 28 FA          525             JR      Z,anaCH0
02C4 DB 88          526             IN      A,(CH0)
02C6 5F             527             LD      E,A
02C7                528
02C7 D3 89          529             OUT     (CH1),A
02C9 DB 81          530     anaCH1: IN      A,(PORTB)
02CB CB 7F          531             BIT     7,A
02CD 28 FA          532             JR      Z,anaCH1
02CF DB 89          533             IN      A,(CH1)
02D1 67             534             LD      H,A
02D2                535
02D2 D3 8A          536             OUT     (CH2),A
02D4 DB 81          537     anaCH2: IN      A,(PORTB)
02D6 CB 7F          538             BIT     7,A
02D8 28 FA          539             JR      Z,anaCH2
02DA DB 8A          540             IN      A,(CH2)
02DC 6F             541             LD      L,A
02DD                542
02DD C3 18 01       543             JP      analog
02E0                544
02E0                545     ;
02E0                546     ; Send to X68000 Analog Data Routine
02E0                547     ;
02E0 F3             548     SEND:   DI
02E1 08             549             EX      AF,AF'
02E2                550
02E2 3E 40          551             LD      A,40H
02E4 3D             552     SLOOP:  DEC     A
02E5 20 FD          553             JR      NZ,SLOOP
02E7                554
02E7 7A             555             LD      A,D
02E8 CB 1F          556             RR      A
02EA CB 1F          557             RR      A
02EC CB 1F          558             RR      A
02EE CB 1F          559             RR      A
02F0 CB E7          560             SET     4,A
02F2 CB EF          561             SET     5,A             ;XHL   = Hi
02F4 D3 82          562             OUT     (PORTC),A
02F6 E6 CF          563             AND     0CFH            ;XHL   = Low
02F8 D3 82          564             OUT     (PORTC),A
02FA                565
02FA DB 80          566             IN      A,(PORTA)
02FC CB 1F          567             RR      A
02FE CB 1F          568             RR      A
0300 CB 1F          569             RR      A
0302 CB 1F          570             RR      A
0304                571
0304 CB A7          572             RES     4,A
0306 CB EF          573             SET     5,A             ;XACK = Hi
0308 D3 82          574             OUT     (PORTC),A
030A EE 30          575             XOR     30H             ;XHL  = Hi
030C D3 82          576             OUT     (PORTC),A       ;2nd Data
030E                577
030E 7B             578             LD      A,E
030F CB 1F          579             RR      A
0311 CB 1F          580             RR      A
0313 CB 1F          581             RR      A
0315 CB 1F          582             RR      A
0317 CB E7          583             SET     4,A
0319 CB EF          584             SET     5,A             ;XHL  = Hi
031B D3 82          585             OUT     (PORTC),A
031D E6 CF          586             AND     0CFH            ;XHL  = Low
031F D3 82          587             OUT     (PORTC),A       ;3rd Data
0321                588
0321 7C             589             LD      A,H
0322 CB 1F          590             RR      A
0324 CB 1F          591             RR      A
0326 CB 1F          592             RR      A
0328 CB 1F          593             RR      A
032A                594
032A CB A7          595             RES     4,A
032C CB EF          596             SET     5,A             ;XACL = Hi
032E D3 82          597             OUT     (PORTC),A
0330 EE 30          598             XOR     30H             ;XHL  = Hi
0332 D3 82          599             OUT     (PORTC),A       ;4th Data
0334                600
0334 7D             601             LD      A,L
0335 CB 1F          602             RR      A
0337 CB 1F          603             RR      A
0339 CB 1F          604             RR      A
033B CB 1F          605             RR      A
033D CB E7          606             SET     4,A
033F CB EF          607             SET     5,A             ;XHL = Hi
0341 D3 82          608             OUT     (PORTC),A
0343 E6 CF          609             AND     0CFH            ;XHL  = Low
0345 D3 82          610             OUT     (PORTC),A       ;5th Data
0347                611
0347 3E 00          612             LD      A,0
0349 CB 1F          613             RR      A
034B CB 1F          614             RR      A
034D CB 1F          615             RR      A
034F CB 1F          616             RR      A
0351                617
0351 CB A7          618             RES     4,A
0353 CB EF          619             SET     5,A
0355 D3 82          620             OUT     (PORTC),A
0357 EE 30          621             XOR     30H             ;XHL  = Hi
0359 D3 82          622             OUT     (PORTC),A       ;6th Data
035B                623
035B CB 1F          624             RR      A
035D CB 1F          625             RR      A
035F CB 1F          626             RR      A
0361 CB 1F          627             RR      A                       ;Dummy
 Work
0363 7B             628             LD      A,E
0364 CB E7          629             SET     4,A
0366 CB EF          630             SET     5,A             ;XHL = Hi
0368 D3 82          631             OUT     (PORTC),A
036A E6 CF          632             AND     0CFH            ;XHL = Low
036C D3 82          633             OUT     (PORTC),A       ;7th Data
036E                634
036E CB 1F          635             RR      A                       ;Dummy
 Work
0370 CB 1F          636             RR      A
0372 CB 1F          637             RR      A
0374 CB 1F          638             RR      A
0376 7C             639             LD      A,H
0377 CB A7          640             RES     4,A             ;XHL = Low
0379 CB EF          641             SET     5,A
037B D3 82          642             OUT     (PORTC),A
037D EE 30          643             XOR     30H             ;XHL = Hi
037F D3 82          644             OUT     (PORTC),A       ;8th Data
0381                645
0381 CB 1F          646             RR      A
0383 CB 1F          647             RR      A
0385 CB 1F          648             RR      A
0387 CB 1F          649             RR      A                       ;Dummy
 Work
0389 7D             650             LD      A,L
038A CB E7          651             SET     4,A
038C CB EF          652             SET     5,A             ;XHL = Hi
038E D3 82          653             OUT     (PORTC),A
0390 E6 CF          654             AND     0CFH            ;XHL = Low
0392 D3 82          655             OUT     (PORTC),A       ;9th Data
0394                656
0394 CB 1F          657             RR      A
0396 CB 1F          658             RR      A
0398 CB 1F          659             RR      A
039A CB 1F          660             RR      A                       ;Dummy
 Work
039C 3E 20          661             LD      A,20H           ;XHL = Low
039E CB A7          662             RES     4,A
03A0 CB EF          663             SET     5,A
03A2 D3 82          664             OUT     (PORTC),A
03A4 EE 30          665             XOR     30H             ;XHL = Hi
03A6 D3 82          666             OUT     (PORTC),A       ;10th Data
03A8                667
03A8 CB 1F          668             RR      A
03AA CB 1F          669             RR      A
03AC CB 1F          670             RR      A
03AE CB 1F          671             RR      A                       ;Dummy
 Work
03B0 7A             672             LD      A,D
03B1 CB E7          673             SET     4,A
03B3 CB EF          674             SET     5,A             ;XHL = Hi
03B5 D3 82          675             OUT     (PORTC),A
03B7 E6 CF          676             AND     0CFH            ;XHL = Low
03B9 D3 82          677             OUT     (PORTC),A       ;11th Data
03BB                678
03BB CB 1F          679             RR      A
03BD CB 1F          680             RR      A
03BF CB 1F          681             RR      A
03C1 CB 1F          682             RR      A                       ;Dummy
 Work
03C3 3E 2F          683             LD      A,2FH           ;XHL = Low
03C5 D3 82          684             OUT     (PORTC),A
03C7 EE 30          685             XOR     30H             ;XHL = Hi
03C9 D3 82          686             OUT     (PORTC),A       ;12th Data
03CB                687
03CB CB 1F          688             RR      A
03CD CB 1F          689             RR      A
03CF CB 1F          690             RR      A
03D1 CB 1F          691             RR      A
03D3 CB 1F          692             RR      A               ;Dummy Work
03D5                693
03D5 3E 20          694             LD      A,20H
03D7 D3 82          695             OUT     (PORTC),A
03D9                696
03D9 DB 81          697             IN      A,(PORTB)
03DB CB 6F          698             BIT     5,A
03DD C2 4A 00       699             JP      NZ,digital
03E0                700
03E0 08             701             EX      AF,AF'
03E1 33             702             INC     SP
03E2 33             703             INC     SP
03E3 FB             704             EI
03E4 C3 18 01       705             JP      analog          ;Instead of RE
TI
03E7                706                                     ;Because of wa
nt of RAM
```

BASIC

カード型データベース(2)

Izumi Daisuke 泉 大介

X-BASICの総集編として取り上げたカード型データベース。前回はデータ構造の基本設計ができたところですが，データベースというからには検索やソートといった機能が必要ですね。これらの機能をcommand関数として付け加えていきましょう。

先月はカード型データベースのできるだけ簡単な仕様を決め，それをプログラムする過程をお届けしたわけですが，いかがだったでしょうか。データがディスク内にどのように保存されるのかおわかりいただけましたか。

今月はカード型データベースの完成編です。今月まわしにしたcommands関数の説明をし，カードの検索，ソートを付け加えていきます。

メニュー選択ルーチン

先月プログラムを入力して動かしてみた方はおわかりかと思いますが，このカード型データベースは最下行に，

1/3：Edit, Del, Clear, Quit

というメニューを表示し，その先頭のEDCQの文字をキーボードから入力することで処理が行われます。これはcommands関数の仕事です。ではその内容を見ていくことにしましょう。

プログラムはコンパイルすることを考えて若干の変更が加えてあります。行末に「追加」「変更」のコメントがある行は先月のものと異なっていますので注意してください。また，先月掲載したinputData関数内のline変数は，コンパイル時にエラーとなってしまいます[1]。変数名をlnに変更してください。line$変数のほうはそのままでOKです。では先頭から見ていきましょう。

1） これはX-BASICのグラフィック関数lineと同じ名前であるためです。X-BASICでは動くので，うっかりそのままにしてしまいました。

2つの大域変数を追加しました。これらは今月追加した検索，ソート機能で使用します。それぞれを説明するときに具体的にどう使っているのかを紹介することにします。

commands関数は4690行から始まります。最初は変数の宣言です。cardpはcards配列内の位置を表すのに使用します。usingCardsは現在使用しているカードの総数，chosen，modeは今月追加した変数で，それぞれcards内に収められているカード数，検索モード保持用に使用します。

chosen変数の追加には疑問を持たれた方もいらっしゃるでしょう。これはcards配列の役割に関係があります。先月は使用されているカードがすべてcards配列に収められていましたが，本来これは選択されたカードだけを収める目的で用意した配列です。検索をかけるとき，検索対象となるカードはcards配列に登録されているものだけです。したがって検索条件をだんだんと厳しくしていけば，cards配列に残るカードはどんどん少なくなっていきます。

ソートするときも同様で，並び替えられるカードはcards配列に登録されているものだけが対象となります。そこで現在cards配列にいくつのカードが収められているのかを保持する変数を用意したというわけです。

commands関数は最初に画面を32行モードにしたら，続いてcollectCards関数を呼び出して使用中のカードをすべてcards配列にセットします。これが初期状態です。collectCards関数が返した使用カード数はusingCardsにセットされ，chosenにもこれが代入されます。最初に表示するのはcards配列の先頭に入っているカードですからcardpは0となり，readCard関数でcards(cardp)のカード，すなわちcards(0)のカードが表示されます。ここまでが初期設定です。

続いてメニューの表示，その選択，機能の実行というループが続くことになります。メニュー表示は若干変更してあり，

0/3 (5)：Edit, ……

という形になっています。最初の数字はcards配列中のどこにいるのかを表しています。次の数字はcards配列に登録されているカードの総数です。そして最後のカッコ内の数字は使用中のカードの総数を意味しています。メニューに表示される機能は以下のとおりです。

Edit	カードの修正
Del	表示中のカードを削除
Clear	cards配列の初期化
eXclude	表示中のカードをcards配列から外す
Find	カードの検索
Sort	カードのソート
Quit	データベースの終了

いずれも大文字になっている英字を入力すれば動作します。

これをinkey$で入力してもらい，switchで処理を分ければメニュールーチンは終了です。簡単ですね。ではメニューごとに分けて見ていくことにしましょう。

各機能の解説

●Edit：カードのエディット

カードのエディットはinputData関数で行います。まずカーソルを（0，0）に移動していますが，これはコンパイル後の動作がX-BASICと異なっていたためです。カーソルが32行目にあるときに画面を31行モードにすると，画面が1行スクロールアップしてしまうのです。画面モードを変更しデータの入力が終わったら，再び画面を32行モードに戻します。ここで，エディットしたカードがすでに使用されているカードかどうかをチェックします。未使用のカードなら（新しいデータを入力したのなら）newCard 関数によって新しいカードをひとつもらい，カード数を変更。最後にwriteCard関数でカードをディスクにセーブします。

●ROLL UP，ROLL DOWN：カードのスクロール

これはcardpを増減し，それに応じたカードをディスクから読み出して画面に表示するだけです。先月はusingCardsでROLL UP時のチェックを行っていましたが，ここはchosen以上にスクロールできないよう変更を加えてあります。

●Del：表示中のカードを削除

先月，カードのディスク上での保存方法を説明しましたが，カードの削除はこれと密接に関係していますので処理が少々面倒です。dbasep配列は実際にデータが収められている位置を示していますから，まずはこれをクリアしなければなりません。対応するusedRecsに0を入れると同時に，dbasepの該当位置を－1にしてデータをクリアします[2)]。

続いて全部－1となったdbasepを～.ratファイルに保存します。これでこのカードはデータをまったく持たない新しいカードとなりました。usedCardsの該当位置のほうも0にして，カードを未使用状態に戻します。最後にusedRecsとusedCardsをratファイルに書き出せば削除作業は終了です。

残る仕事は削除したカードをcards配列から外すことです。cardpから最後のカードまでを順に繰り上げていき，削除したカードをcards配列から消します。そして次のカードを表示，usingCards，chosenを更新します。

●Clear：cards配列の初期化

先ほども説明したように，cards配列は「選択されたカード」を収めます。検索を繰り返していくうちにcards配列に残るカードはどんどん少なくなっていくことでしょう。これを初期化し，使用されているカードをすべてcards配列に収め直すのがこのコマンドです。commands関数の先頭でやったのと同じように，collectCards関数を呼び出して処理します。

●Quit：データベースの終了

コマンド入力ループはflag変数が1である間回り続けます。flag変数に0を代入すればwhile～endwhileループは終了となります。

●eXclude：表示中のカードをcards配列から外す

ここからが，今月新たに追加したコマンド群です。検索，ソートの対象となるcards配列中のカードですが，なかには明らかに意に沿わないため条件を満たしてはいるが対象から外そうというものも出てきます。2コンの重さはどうにも我慢できないからヤダとか，ペン太ックスはホールディングがいまいち好みではないから外そうとか。Xコマンドはこの処理を行います。

プログラムはDelコマンドを簡略化した形になっています。cards配列から現在表示しているcardp番目のカードを取り除き，残りのカードを詰めるだけです。

●Find：カードの検索

さあ，本日の目玉商品が登場しました。ここまで何度もいってきたように，これはcards配列に登録されているカードから，条件に合うカードを見つけ出してcards配列に再セットしようというコマンドです。でも，本当にcards配列の中から探すだけでいいのでしょうか。「重量が600g未満で幅が155mm未満のカメラ！」と指定して探した結果に満足できず，「では重量が600g未満ならよしとしよう」と条件を緩めた検索結果を加えたい，あるいは「多重露出ができれば重量も幅もどうでもいい」と思いっ切り譲歩して検索した結果も合わせて見たいという要求はないのでしょうか。

これを解決するため，検索に2つのモードを設けることにしました。ひとつはノーマル検索で，これはcards配列中のカードから条件に合致するものを探します。もうひとつは追加検索でこちらはcards配列に登録されたカード以外のものが検索対象となります。Fキーが押されると画面最下行はこの検索モードの入力となります。

Normal, Additional >

と表示し，nかa，あるいはESCキーが押されるまでループ。キー入力を変数modeに格納します。ESCキーが押された場合は検索中断ということで処理終了です。

続いて検索条件の入力です。これはgetKey関数で行います。getKey関数を見てください。条件の入力はinputData関数を使います。カードへのデータ入力やカードの設計に使っているアレですね。入力されたデータをData配列に収めるのがinputData関数の役目でした。

プログラムはまずData配列をクリアし，条件入力のループに入ります。画面をクリアしたあとData配列になにか入っていたらその行を画面の該当位置に表示するのですが，最初は当然なにも入っていませんので表示なしです。31行モードにしてinputData関数を呼び，条件を入力してもらいます。そう，inputData関数で入力するのですから，カードにデータを入れるのと同じ方法で条件も指定するのです。具体的に説明しましょう。

2）実際にdatファイルに収められている文字データのほうをクリアする必要はありません。どのカードからも参照されなくなったということはクリアしたというのと同じですからね。

3) 重量600g未満で幅155mm未満のカメラ，というのはANDで検索することになります。

4) 重量600g未満か多重露出ができるカメラ，というのはOR検索です。

重量　　　#

と設計されている場所があれば，データ入力時に，

重量　　　■

とカーソルが#の場所で点滅し，カメラの重量を入力できるのは先月説明したとおりです。同じinput Data関数を使っているのですから，条件の入力時にもカーソルはここで点滅します。もし600gより軽いカメラを探したいのなら，ここで，

重量　　　<600

と入力します。これは「重量が600より小さい」と読めますね。また，多重露出のできるカメラを探したいなら，

多重露出　　=○

とでもなるでしょう。もちろんデータが，

多重露出　　○

あるいは，

多重露出　　×

という形で有無を区別している場合の話ですが。

このように検索条件は，

「演算子」＋「比較データ」

という形で与えます。使える演算子は，

= 比較データと等しい
< 比較データより小さい
> 比較データより大きい
! 比較データと等しくない
) 比較データが含まれている
(比較データに含まれている

の6つです。>=などと組み合わせて使うことはできません。悪しからず。

検索条件の入力が終了したら，Data配列を調べて条件がちゃんと与えられているかどうかを調べます。Data配列の中でデータの入っているものは，先頭が必ずこれら6つの演算子でなければなりませんのでそれをチェックするのです。間違いがあればその行をシアンで表示し，入力のやり直しです。

では，commands関数に戻りましょう。getKey関数で検索条件の入力が終了したら検索の開始……，いえいえ，検索にはもうひとつ考えておかなければならないことがあります。それは，列挙した複数の条件をすべて満たすものを探す（ANDで検索する[3]）のか，ひとつでも満たしていればOKとする（ORで検索する[4]）のかです。そこで最後に検索方法の入力メニューを表示します。これも画面最下行に，

検索方法：And, Or >

と表示して入力してもらいます。ここでもESCで中断できるようにしておくのがいいでしょう。入力してもらった検索方法は，最初に出たconditionという大域変数に0か1を代入して保存しておきます。検索を行う関数はこの変数を参照し，検索方法を決定します。

前準備が整ったら検索開始です。ノーマル検索の場合はinCards関数で，追加検索の場合はaddCards関数で検索を行います。いずれの関数も検索終了時にcards配列に抽出されたカードの総数を返しますので，これをchosen変数に代入してやります。cardpのほうはノーマル検索時にはcards配列の先頭に，追加検索時には追加されたカードの先頭になります。

ではinCards関数です。cards配列内のカードを順にfindCards関数に渡し，そのカードが検索条件に合致するかどうかを判定します。合致すればOKですが，合致しなければそのカードはcards配列から削除されます。

addCards関数のほうはcards配列の代わりにusedCardsから使用されているカードを探し出しfindCards関数に渡します。ただしcards配列に登録されているものは判定対象とはしないというのが仕様ですから，ちょっとした細工が必要です。usedCards配列はカードの使用状況を1と0で区別しています。usedCards(n)=1ならn番目のカードは使われているという印です。ここでcards配列に登録されているカードに対応する数値を2に置き換えてやれば，usedCards配列中で1になっているカードは「使用されていてcards配列には入っていない」カードとなりますね。ああ，簡単。これをfindCards関数に渡してチェックしてもらうだけです。条件に合致すればそのカードをcards配列に加えていきます。最後にusedCards配列の値を1に戻して終了です。

カード検索のトリを務めるのはfindCards関数です。再びratファイルやdatファイルが登場しますので注意してください。findCards関数は引数として条件と比較するカードの番号を受け取ります。まず最初にやることは，渡されたカードのデータ情報をratファイルからdbasep配列に取り出すことです。続いてData配列を調べ，条件が設定されている行があれば演算子をop変数に，比較データをcond変数にセットし，対応する行のデータをdatファイルからline$変数に取り出して吟味を始めます。リストでは7180行からが演算子によって処理を分けているところです。すべて文字列の比較で処理している点に注意してください。

こうしてData配列中の条件を順に調べていくのですが，検索方法によって検索打ち切りの条件が異

なります。AND検索（condition＝1）なら設定されている条件がすべて満たされなければなりませんから，不成立の条件があればそこで打ち切りです。逆にOR検索（condition＝0）なら，ひとつでも条件が成立すればそこで打ち切っていいわけです。条件成立かどうかを返してfindCards関数は終了です。

Sort：カードのソート

commands関数もいよいよ最後の項目になりました。カードのソートです。ここでもgetKey関数を使ってソート条件を入力します。こちらの条件は実に単純です。カメラの重量を昇順にソートしたければ，

　　重量　　　<

と書くだけです。逆に降順にソートしたければ'>'を使います。

　　card m < card n < card o

という意味だと考えればわかりやすいでしょう。プログラムはgetKey関数で入力されたData配列を調べ，条件が設定されている行を見つけたところでソートを開始します。最初に見つけた場所だけが有効で，複数キーによるソートには対応していません。

実際にソートを行うのはshellSort関数です。その名のとおりシェルソートというアルゴリズムを使ってソートを実行します。ではshellSort関数を見てみましょう。shellSort関数はcards配列中のデータ数，ソートのキーとなる行，そして昇順・降順の区別を引数として受け取ります。まず最初にやることは，cards配列に登録されているカードのソートキーとなる行をdatファイルから取り出し今月追加したkeys配列にセットすることです。キーとなる行は何度も参照されます。そのたびにファイルから読み込んでいたのでは時間がかかり過ぎるだろうと考えて最初に読み込んでおくことにしました。

キーの読み込みが終わったらいよいよシェルソートの開始です。シェルソートはちょっと変わったソートの仕方をします。適当な大きさのgapを考え，i番目のデータとi＋gap番目のデータを比較して，大小関係が逆なら入れ替えを行うのです。最初gapはデータ数の半分に設定され，データ列の前半と後半で入れ替えが行われます。次にgapはさらに半分にされ，半分にされ……，ついにgapが0になったときにはデータは綺麗に整列しているという方法です。データのバラツキをまずはザッとならし，次第に細かく整えていくというイメージです[5]。

まぁ，今回はソートの特集ではないので深入りするのは止めましょう。shellSort関数では，データの入れ替えが必要となった時点でkeys配列のキーと，それに対応するcards配列のカード番号を同時に入れ替えています。

さぁて、来月の調理実習は〜

X-BASIC総集編のカード型データベースはいかがでしたか？　なに，inputData関数が遅すぎる？　ごもっとも。本文でも触れているように，このプログラムはXCでコンパイルすることができます。速度に不満のある方はコンパイルして利用してみてください。コンパイルは簡単です。XCのシステムディスクを起動して本プログラムの入ったディスクをBドライブにセット。

　　b:

で画面に「B>」と表示されたら，

　　cc database.bas

とするだけであとは自動的に終了します。実行は，

　　database

と入力すればOK。ぜひ挑戦してみてください[6]。

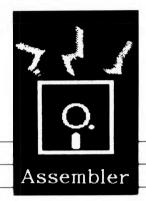
さて，来月は本プログラムをより便利に使うためのツールをいくつか用意し，最後を締めくくりたいと思います。ではまた，来月。

5）これではなんの説明にもなっていませんね。バブルソートなどと比較するとわかりやすいと思うのですが……，なんせページ数がないもので。ごめんなさい，皆さんの努力に期待します。

6）XC ver.2.0でコンパイルすると，山のようにWarningが出ますが，無視して差し支えありません。厳しくなったチェックにBCが対応していないのが原因です。

リスト1　カード型データベース（その2）

```
 150 int condition                     /* 検索方法
 160 str keys(100)[97]                 /* ソート用キー配列
4660 /*
4670 /* メインメニュー&カード閲覧
4680 /*
4690 func commands()
4700   str ch
4710   int flag = 1
4720   int cardp, usingCards
4730   int chosen                      /* cards内のカード数
4740   str mode                        /* 検索モード
4750   int i
4760   /*
4770   console 0,32,0                  /* 画面を32行モードに変更
4780   usingCards = collectCards()     /* 使用中のカード数を数え
4790   chosen = usingCards             /* 「追加」chosenにセット
4800   cardp = 0                       /* 最初のカードを
4810   readCard( cards( cardp ))       /*  表示する
4820   while flag
4830     locate 0, 31                  /* 最下段にメニューを表示
4840     print cardp;"/";chosen;"(";usingCards;")"; /* 「変更」
4850     print " : Edit, Del, Clear, ";  /* 「変更」
4860     print "eXclude, Find, Sort, Quit > "; /* 「追加」
4870     ch = inkey$
4880     switch asc( ch )
4890       case 'e'                    /* カードのエディット
4900       case 'E'
4910         locate 0, 0               /* 「追加」一旦カーソルを移動
4920         console 0,31,1            /* 31行モードで
4930         inputData( 1 )            /* データの入力
4940         console 0,32,0
4950         if cards( cardp ) = 255 then {
4960           cards( cardp ) = newCard()
4970           if usingCards < 99 then { /* 「変更」
4980             usingCards = usingCards + 1
4990             chosen = chosen + 1
5000           }
5010         }
5020         writeCard( cards( cardp )) /* カードのセーブ
5030       case 14                     /* ROLL UP
5040         if cardp = chosen then break /* 「変更」
5050         cardp = cardp + 1         /* 次のカードを
5060         readCard( cards( cardp ))  /*  読み込む
5070         break
5080       case 15                     /* ROLL DOWN
5090         if cardp = 0 then break
5100         cardp = cardp - 1         /* 前のカードを
5110         readCard( cards( cardp ))  /*  読み込む
5120         break
5130       case 'd'                    /* 表示中のカード削除
5140       case 'D'
5150         if cards( cardp ) = 255 then break
5160         for i=0 to 30             /* レコード解放
5170           if dbasep( i ) <> -1 then {
5180             usedRecs( dbasep( i )) = 0
5190             dbasep( i ) = -1
5200           }
5210         next
5220         fseek( rat, 2100 + cards( cardp )*4*31, 0 )
5230         fwrite( dbasep, 31, rat )  /* ディスク上のdbasep更新
5240         /*
5250         usedCards( cards( cardp )) = 0 /* カード解放
5260         fseek( rat, 0, 0 )
5270         fwrite( usedCards, 100, rat )  /* カードと
5280         fwrite( usedRecs, 2000, rat )  /*  レコードの更新
5290         for i=cardp to 98         /* カードを削除
5300           cards( i ) = cards( i+1 )
5310         next
5320         cards( 99 ) = 255
5330         readCard( cards( cardp ))  /* 次のカードを読み込む
5340         usingCards = usingCards - 1
5350         chosen = chosen - 1
5360         break
5370       case 'c'                    /* 使用中カードの再整列
5380       case 'C'
5390         usingCards = collectCards()
5400         chosen = usingCards
5410         cardp = 0
5420         readCard( cards( cardp ))
5430         break
5440       case 'q'                    /* データベースの終了
5450       case 'Q'
```

```
5460          flag = 0
5470          break
5480        /*
5490        /* 以下追加コマンド
5500        /*
5510        case 'x'
5520        case 'X'
5530          if cards( cardp ) = 255 then break
5540          for i=cardp to 98           /* カードをcardsから削除
5550            cards( i ) = cards( i+1 )
5560          next
5570          cards( 99 ) = 255           /* 最後には未使用フラグを
5580          chosen = chosen - 1
5590          readCard( cards( cardp ))  /* 次のカードを読み込む
5600          break
5610        case 'f'                      /* カードの検索
5620        case 'F'
5630          repeat                      /* 検索モード入力
5640            locate 0, 31
5650            print chr$(5);"Normal, Additional > ";
5660            mode = inkey$
5670          until instr( 1, "nNaA"+chr$(27), mode )
5680          if mode = chr$( 27 ) then break
5690          getKey()                    /* 検索条件入力
5700          repeat                      /* 検索方法入力
5710            locate 0, 31
5720            print chr$(5);"検索方法：And, Or > ";
5730            ch = inkey$
5740          until instr( 1, "aAoO"+chr$(27), ch )
5750          if ch = chr$(27) then break
5760          if toupper( asc( ch )) = 'A' then {
5770            condition = 0             /* 検索方法を保存
5780          } else {                    /*  inCards、addCards関数で
5790            condition = 1             /*   参照される
5800          }
5810          switch asc( mode )
5820            case 'n'                  /* 通常検索
5830            case 'N'
5840              chosen = inCards( chosen ) /* 合致カード数を更新
5850              cardp = 0               /* cardpは先頭へ
5860              break
5870            case 'a'                  /* 追加検索
5880            case 'A'
5890              cardp = chosen          /* cardpは追加部へ
5900              chosen = addCards( chosen ) /* 合致カード数を更新
5910              break
5920            default
5930              break
5940          endswitch
5950          readCard( cards( cardp ))  /* カードを表示
5960          break
5970        case 's'                      /* カードのソート
5980        case 'S'
5990          getKey()                    /* ソート条件入力
6000          for i=0 to 30
6010            if Data( i ) <> "" then { /* 条件があればソート
6020              shellSort( chosen-1, i, left$( Data( i ), 1 ))
6030              break
6040            }
6050          next
6060          cardp = 0                   /* cardpは先頭へ
6070          readCard( cards( 0 ))      /* カードを表示
6080          break
6090        default
6100          break
6110      endswitch
6120    endwhile
6130    locate 0, 0                       /* 「追加」一旦カーソルを移動
6140    console 0,31,1                    /* 31行モードに
6150  endfunc
6160  /*
6170  /* 検索用キー入力
6180  /*
6190  func getKey()
6200    int i, flag
6210    str op                            /* 演算子を保持
6220    for i=0 to 30
6230      Data( i ) = ""                  /* Dataをクリア
6240    next
6250    repeat
6260      cls
6270      for i=0 to 30
6280        if Data( i ) <> "" then {     /* データがあれば表示
6290          locate dataEntry( i ), i
6300          print Data( i );
6310        }
6320      next
6330      locate 0, 0                     /* 一旦カーソルを移動
6340      console 0, 31, 1                /* 31行モードに
6350      inputData( 1 )                  /* データを入力
6360      console 0, 32, 0                /* 32行に戻す
6370      /*
6380      cls
6390      flag = 0
6400      for i=0 to 30                   /* 入力キーのチェック
6410        if Data( i ) = "" then continue
6420        op = left$( Data( i ), 1 )
6430        if instr( 1, "=<>!()", op ) then {
6440          color 3                     /* 条件がOKなら白で
6450        } else {
6460          flag = 1
6470          color 1                     /* さもなければシアンで
6480        }
6490        locate dataEntry( i ), i
6500        print Data( i );              /* データを表示する
6510      next
6520      if flag then {                  /* 間違いがあれば
6530        color 1                       /*  シアンで指摘
6540        locate 0, 31
6550        print "条件指定に間違いがあります ";
6560        op = inkey$
6570      }
6580      color 3
6590    until flag = 0                    /* 間違いがなくなるまで
6600  endfunc
6610  /*
6620  /* cardsの中から条件に合致するものを調べる
6630  /*
6640  func int inCards( maxNum )
6650    int i, j
6660    i = 0
6670    while i < maxNum                  /* cards内を順に検索
6680      if findCards( cards( i )) = 0 then { /* 合致しなければ
6690        for j=i to maxNum-1               /*  そのデータを削除
6700          cards( j ) = cards( j+1 )
6710        next
6720        cards( maxNum ) = 255         /* 最後のカードを未使用にし
6730        maxNum = maxNum - 1           /* 総数を減じる
6740        continue                      /* 削除時はiを更新しない
6750      }
6760      i = i + 1
6770    endwhile
6780    return( maxNum )                  /* 総数を返す
6790  endfunc
6800  /*
6810  /* cards以外のカードを調べる
6820  /*
6830  func int addCards( maxNum )
6840    int i, num
6850    for i=0 to maxNum-1               /* cards内のカードに対応する
6860      usedCards( cards( i )) = 2      /* usedCardsを2にする
6870    next
6880    num = maxNum
6890    for i=0 to 99                     /* 0番のカードから順に
6900      if usedCards( i ) = 1 then {    /* 使用されているカードで
6910        if findCards( i ) then {      /* 条件合致をチェック
6920          cards( num ) = i            /* 合致すればcardsに登録
6930          num = num + 1               /* 総数を増やす
6940        }
6950      }
6960    next
6970    for i=0 to maxNum-1               /* 2にしたものを1に戻す
6980      usedCards( cards( i )) = 1
6990    next
7000    return( num )                     /* 総数を返す
7010  endfunc
7020  /*
7030  /* カードが条件に合致するか調べる
7040  /*
7050  func int findCards( No )         /* Noのカードが合致するか調べる
7060    int flag
7070    int i
7080    str op, cond[97]                  /* condは演算子以降を保持
7090    fseek( rat, 2100 + No*4*31, 0 )   /* Noまでシークし
7100    fread( dbasep, 31, rat )          /* カードを読み込む
7110    for i=0 to 30
7120      if Data( i ) <> "" then {       /* 条件が設定されているなら
7130        flag = 0
7140        op = left$( Data( i ), 1 )     /* 演算子と
7150        cond = mid$( Data( i ), 2, 96 ) /* 比較データを取り出し
7160        fseek( dbase, dbasep(i)*97, 0 )
7170        freads( line$, dbase )        /* カードの該当行を取り出して
7180        switch asc( op )              /* 演算子によって分ける
7190          case '='                    /* 等しい
7200            if line$ = cond then flag = 1
7210            break
7220          case '<'                    /* 小さい
7230            if line$ < cond then flag = 1
7240            break
7250          case '>'                    /* 大きい
7260            if line$ > cond then flag = 1
7270            break
7280          case '!'                    /* 等しくない
7290            if line$ <> cond then flag = 1
7300            break
7310          case '('                    /* 条件に含まれる
7320            if instr( 1, cond, line$ ) then flag = 1
7330            break
7340          case ')'                    /* 条件が含まれる
7350            if instr( 1, line$, cond ) then flag = 1
7360            break
7370        endswitch
7380        if condition = 0 then {       /* AND検索で
7390          if flag = 0 then {          /* 条件不成立なら
7400            break                     /* 検索終了
7410          }
7420        } else {                      /* OR検索で
7430          if flag = 1 then {          /* 条件成立なら
7440            break                     /* 検索終了
7450          }
7460        }
7470      }
7480    next
7490    return( flag )                    /* 見つかったかどうかを返す
7500  endfunc
7510  /*
7520  /* シェルソート
7530  /*
7540  func shellSort( num, keyLine, sortKey;str )
7550    int gap, i, j, keyln( 0 )
7560    int tmp                           /* swap用
7570    /*
7580    for i=0 to num                    /* sortキーの読み込み
7590      fseek( rat, 2100 + cards( i )*4*31 + keyLine*4, 0 )
7600      fread( keyln, 1, rat )
7610      fseek( dbase, keyln( 0 )*97, 0 )
7620      freads( line$, dbase )
7630      keys( i ) = line$
7640    next
7650    /*
7660    gap = num / 2                     /* シェルソート開始
7670    locate 0, 31                      /* 最下行に
7680    while gap > 0
7690      print gap;"->";                 /* gapを表示（気休め）
7700      for i= gap to num
7710        j = i - gap                   /* jはiとgapだけ離れている
7720        while j >= 0
7730          if sortKey = ">" then {     /* 並べ変える必要のない場合
7740            if keys( j ) >= keys( j+gap ) then break
7750          } else {                    /* をここで排除する
7760            if keys( j ) <= keys( j+gap ) then break
7770          }
7780          tmp = cards( j )            /* カードの並べ変え
7790          cards( j ) = cards( j+gap )
7800          cards( j+gap ) = tmp
7810          line$ = keys( j )           /* キーの並べ変え
7820          keys( j ) = keys( j+gap )
7830          keys( j+gap ) = line$
7840          j = j - gap
7850        endwhile
7860      next
7870      gap = gap / 2
7880    endwhile
7890  endfunc
```

C,X-BASICの関数を作成する

Murata Toshiyuki 村田 敏幸

グラフィック関係のサブルーチンを作ってきましたが，これらはC言語やX-BASICのプログラムからも呼び出して使用できるとたいへん便利です。今回はCの関数，X-BASICの外部関数をマシン語で書く場合の方法を解説します。

ここ3回にわたって，グラフィック周りのプリミティブなサブルーチンをいくつか作ってきた。課題扱いだったものも含めてリストで示したものを数え上げると結構な数にのぼる。今月は，これらサブルーチンをC言語，X-BASICから利用する場合を例に，Cの関数をマシン語で書く方法，X-BASICの外部関数の作成方法を見ていこうと思う。

変数の扱い方

予備知識として，Cにおける変数の扱いについてマシン語の立場から眺めてみる。

まず，データ型と記憶領域の大きさの関係を示そう。XCにおける基本データ型と占める領域は，

char	1バイト
short	2バイト
int	4バイト
long	4バイト
float	4バイト
double	8バイト

である。また，ポインタは，それが何を指していようと常に4バイトだ[1]。さらに，配列は同じ型のデータが連続したメモリに要素数だけ並んだもの[2]であり，要素のサイズ×要素数分の領域を占める。

構造体はいくつかのデータ型を寄せ集めてまとめて扱えるようにしたもの[3]だから，基本的には構造体の各メンバーのサイズを合計しただけの大きさを持つ。ただし，68000では偶数境界をまたいだワード/ロングワードのアクセスが禁止されている関係で，char型のメンバーの直後に1バイトの詰め物が入る場合がある。

さて，Cの変数はその記憶場所がどこにとられるかによって次のように分類される。

1) メモリ上の固定領域にとられるもの（長命：プログラムの実行中ずっと生き続ける）
2) スタック上にとられるもの（短命）
3) レジスタ上にとられるもの（短命）

関数の外部で定義された変数と，関数の内部でとくに静的（static）であると明示された変数は1)，関数内の自動的（auto）な変数は2)，そして，いわゆるレジスタ変数として宣言されたもの（register）は空いているレジスタがあれば3)，そうでなければ2）のパターンとなる。リスト1に示した例では，

a，b，c，d，g，hは	1)
e，x，yは	2)
fは	3)

にそれぞれ該当する。Cでは関数への引数の引き渡しにスタックを利用するので，関数fooの仮引数であるx，yも2)のパターンに含まれることに注目してもらいたい。

マシン語に当てはめて考えると，1)はデータセクションやbss（Block Storage Section：ブロックストレージセクション）[4]に確保したワークエリア，2)はスタック上に確保したワークエリアと，サブルーチン呼び出し時にスタックに積まれた引数，3)はレジスタそのものに相当するわけだ。

通用範囲（scope：スコープ）について分類するなら，次のようになる。

ア） プログラム中のどこからでも参照できるもの
イ） 1本のソースファイル中（の定義された行からファイルの終わりまで）でのみ参照できるもの
ウ） 特定のブロック内（“{”～“}”）でのみ参照できるもの

関数の外部でstaticをつけずに定義するとア)，関数外部でstaticに定義するとイ)，関数内で定義するとウ)になる。リスト1では，

a，bは	ア)
c，dは	イ)
e，f，g，h，x，yは	ウ)

1） アドレスを保持するのがポインタ変数なのだから。
2） 配列は.dcがずらずら並んだもの，もしくは，.dsで確保された一連のメモリ領域をイメージすればよい。
3） 構造体（のテンプレート）は.offsetによるオフセット表のイメージ。
4） storageは英語の発音に従えば，ストリッジないしはストーリッジと表記されるべきなのだろうが，ここではよく見かける表記を採用した。ちなみに，“(データの）保管場所”とか，もう少し意訳すれば“メモリ領域”程度の意味。

5) “__”をつけるのは，コンパイラが自分で勝手に使うラベルとCソース中の識別名がかちあわないようにするため。

6) ここでいうコモンエリアはHuman68k Ver.2.0のconfig.sysのcommon行で指定するものとはまったくの別のもの。念のため。

7) リスト2ではたまたま.bssと.endの間に.commが置かれているが，実際には.commはどこに置いてもかまわない。

8) コモンエリア以外のbssは初期化されることはない(初期値はメモリ上のゴミ)。

9) ただし，staticがついていようといなかろうと，初期値の指定がない静的な変数は0で初期化されるはずだから，このコンパイル結果は誤りだと思う。変数cの場合と同様に，.dc.lで領域を確保するのが正しい(.commで定義するのもまた誤り。外部からも参照可能になってしまい，スコープが定義と違ってしまう)。

10) これも本当は0で初期化すべきものだろう。

にそれぞれ該当する。

AS.X仕様のアセンブリ言語に当てはめてみるとア)は.xdef(ないしは.global)で外部宣言されたワーク，イ)ウ)は.xdefで宣言されていない，もしくはスタック上の一時的なワークと考えることができる。また，リスト1には出てきていないが，C言語のextern宣言は，“プログラムのどこかで同名の変数が定義されている”ということをコンパイラに知らせるものだから，ちょうど.xrefに相当する。

実際にリスト1をXC Ver.2.0でコンパイルしてみた結果がリスト2。XCは馬鹿正直なコードを出してくれるので，このような例にはうってつけだ。リスト1のa～hの変数定義がどのようにコンパイルされているか見てもらおう。

・変数a

“外部からも参照可能なメモリ上のintの変数で初期値は1”だから，データセクション中に.dc.lで領域が確保される(26行)と同時に.globlによる外部定義が行われている(2行)。見てのとおり，XCでは，変数に割り当てられたメモリ領域の先頭アドレスを，“Cにおける識別名の頭にアンダーバー(_)をつけたラベル”で表す[5]。

・変数b

“外部からも参照可能なメモリ上のintの変数で初期値は指定されていない”というパターンだ。Cでは静的な変数は初期値が指定されなければ0で初期化されることになっており，リスト2では.comm疑似命令によってコモン(common：共通)エリア[6]に4バイト分の0で初期化された領域を確保している(35行)。コモンエリアは，bssの一部[7]で，xファイルがメモリに読み込まれるときにHuman68kによって0で初期化される[8]領域だ。

・変数c

“このソース内部でのみ通用するメモリ上のintの変数で初期値は1”だ。28行で.dc.lにより領域は確保されているが，外部定義はない。

・変数d

“このソース内部でのみ通用するメモリ上のintの変数で初期値の指定なし”なので，bssに.ds.bにより4バイトの領域だけ[9]が確保されている(34行)。外部定義もない。

・変数e

関数内で記憶クラスの指定なしに宣言された変数はautoと見なされ，関数呼び出しのたびにスタックフレーム上に領域が一時的に確保される。リスト2では，7行で4バイト分のローカルエリアが確保され，変数eは“−4(a6)”で参照されているのがわかるだろう。

・変数f

レジスタ変数だ。XCではレジスタ変数はd3～d7，a3～a5のいずれかに割り当てられる。通常の変数はデータレジスタに，ポインタはアドレスレジスタに，と決まっているらしい。リスト2では変数fはd7レジスタに割り当てられている。

・変数g

関数内でのみ通用するローカル変数だが，staticがついているから領域はメモリ上の固定領域にとられる。リスト2では37行でbssに4バイトの領域が確保されている[10]。ただ，その領域にふられたラベルが“_g”ではなく，“L5”になっている。これは，もし関数の外側でgという別の変数が定義されていても，ラベル定義がぶつからないようにする

リスト1 VARTEST.C

```
 1: int a = 1;
 2: int b;
 3:
 4: static void foo( x, y )
 5: int x, y;
 6: {
 7:     int e;
 8:     register int f;
 9:     static int g = 1;
10:     static int h;
11:
12:     e = x;
13:     h = y;
14:     f = g;
15: }
16:
17: static int c = 1;
18: static int d;
19:
20: int main()
21: {
22:     foo( c, 50000 );
23:
24:     return 0;
25: }
```

リスト2 VARTEST.S

```
 1:          INCLUDE fefunc.h
 2:          .GLOBL  _a
 3:          .GLOBL  _main
 4:          .XREF   __main
 5:          .TEXT
 6: _foo:
 7:          LINK    A6,#-4
 8:          MOVE.L  D7,-(SP)
 9:          MOVE.L  8(A6),-4(A6)     *e = x;
10:          MOVE.L  12(A6),L5        *h = y;
11:          MOVE.L  L4,D7            *f = g;
12:          MOVE.L  (SP)+,D7
13:          UNLK    A6
14:          RTS
15: _main:
16:          LINK    A6,#0
17:          MOVE.L  #50000,-(SP)     *
18:          MOVE.L  _c,-(SP)         *
19:          JSR     _foo             *foo( c, 50000 );
20:          ADDQ.L  #8,SP            *
21:          MOVEQ.L #0,D0            *return 0;
22:          UNLK    A6               *
23:          RTS                      *
24:          .DATA
25: _a:
26:          .DC.L   $00000001
27: _c:
28:          .DC.L   $00000001
29: L4:
30:          .DC.L   $00000001
31:          .EVEN
32:          .BSS
33: _d:
34:          .DS.B   4
35:          .COMM   _b,4
36: L5:
37:          .DS.B   4
38:          .END
```

ためだ。変数gはあくまでローカル変数だから，ほかの関数から参照されることはない。

・変数h

ローカルでstaticな変数に初期値を与えた例だ。データセクションに.dc.lで領域が確保されている(30行)。

・変数x，y

関数の仮引数だ。auto変数同様，スタックフレーム上の位置で示されている。リスト2では“8(a6)”が変数x，“12(a6)”が変数yにあたる。

・関数foo，main

ついでに、関数にも目を向けてもらいたい。関数fooはstaticに宣言してある。したがって，.xdefによる外部定義は行われず，ほかのソースから参照することはできない。対して，mainは記憶クラスの指定がないので，3行で外部定義が行われている。

余談ながら，4行で外部参照定義された__mainというラベルは，Cのスタートアップルーチンの先頭アドレス，つまりはこのCプログラムの実行開始アドレスを意味している[11]。この定義があることで，リンク時にリンカは__mainというシンボルをライブラリから探し，結果としてスタートアップルーチンをリンクすることになるわけだ。

C言語の関数をマシン語で

そろそろCの関数をアセンブリ言語で書く方法を考えてみよう。上で示した情報から，いくつかの約束ごとが見えてくる。

1)　関数は，Cで使う関数名の頭に“_”をつけたラベル名で表されるサブルーチンとして記述する。

2)　そのサブルーチン名は.xdefまたは.globlで外部定義しておく。

3)　引数の受け渡しはスタックを介して行う。

4)　呼び出し元の関数でレジスタ変数を使っているかもしれないから，サブルーチン内でd3～d7，a3～a5を使うときには値を保存しておく。

5)　当然，フレームポインタであるa6とスタックポインタであるa7 (sp) の値も保存する必要がある。

また，リスト1と2を注意深く見比べると，次のようなこともわかる。

6)　関数へ渡す引数をスタックに積む順序は，Cにおける引数の並び順の逆（リスト2の17～18行）

7)　関数の戻り値はd0レジスタに入れて返す（リスト2の21行）[12]。ただし，fooのようなvoid型の(値を返さない）関数では戻り値の心配をする必要はない。

では，試しに先月のサブルーチンgcopyをCから呼び出せるようにしてみよう。単に呼び出せるようにするだけであれば，非常に簡単だ。先月のリスト10（GCOPY.S）中，4行の，

```
.xdef    gcopy
```

の直後に，

```
.xdef    _gcopy
```

を，また，22行の，

```
gcopy:
```

の直後に，

```
_gcopy:
```

を挿入し，アセンブルし直せば，サブルーチン_gcopy（実体はサブルーチンgcopyと同じもの）がgcopy( )という名前の関数として呼び出せる形式になる。

ただ，サブルーチンgcopyは転送元/転送先の領域の座標を任意のメモリ領域に書き込み，その先頭アドレスをスタックに積んで呼び出すようにできており，これをCで実現するにはリスト3のように構造体に引数を代入して，その構造体の先頭アドレスを関数に渡すしかない。また，gcopy内ではスーパーバイザ空間にアクセスすることになるので，gcopy( )を呼び出す前にあらかじめスーパーバイザモードへ移行しておく必要もある。仕様です，と突っぱねることもできるが，どうせCの関数にするのなら，

11)　Cでは一部の関数の初期化のために，関数mainの実行に先立って，このスタートアップルーチンが実行される。関数mainはその中からサブルーチンコールされることになる。

12)　doubleや構造体を戻り値とする関数ではまた違ってくるのだが，ここでは触れない。

リスト3　SMPL.C

```
 1: int main()
 2: {
 3:    struct GCOPYBUF {
 4:        short  x0, y0;
 5:        short  x1, y1;
 6:        short  x2, y2;
 7:    } gcopybuf;
 8:
 9:            :
10:
11:    gcopybuf.x0 = 0;
12:    gcopybuf.y0 = 0;
13:    gcopybuf.x1 = 50;
14:    gcopybuf.y1 = 50;
15:    gcopybuf.x2 = 100;
16:    gcopybuf.y2 = 0;
17:    gcopy( &gcopybuf );
18:
19:            :
20: }
```

リスト4　_GCOPY.S

```
 1:          .include       iocscall.mac
 2: *
 3:          .xdef   _gcopy
 4:          .xref   gcopy
 5: *
 6:          .offset 8
 7: *
 8: X0:      .ds.l   1
 9: Y0:      .ds.l   1
10: X1:      .ds.l   1
11: Y1:      .ds.l   1
12: X2:      .ds.l   1
13: Y2:      .ds.l   1
14: *
15:          .text
16:          .even
17: *
18: _gcopy:
19:          link    a6,#0
20:
21:          suba.l  a1,a1
22:          IOCS    _B_SUPER
23:
24:          move.w  Y2+2(a6),-(sp)
25:          move.w  X2+2(a6),-(sp)
26:          move.w  Y1+2(a6),-(sp)
27:          move.w  X1+2(a6),-(sp)
28:          move.w  Y0+2(a6),-(sp)
29:          move.w  X0+2(a6),-(sp)
30:
31:          move.l  sp,-(sp)
32:          jsr     gcopy
33:
34:          tst.l   d0
35:          bmi     done
36:
37:          movea.l d0,a1
38:          IOCS    _B_SUPER
39:
40: done:    unlk    a6
41:          rts
42:
43:          .end
```

gcopy(0, 0, 50, 50, 100, 0)；

のような素直な形式で，ユーザーモードからも呼び出せるようにしたいところだ。

そこで，また別の方法を検討する。サブルーチンgcopyとCプログラムとの間にワンクッション入れて，つじつまを合わせることにしよう。リスト4だ。サブルーチンの頭でスーパーバイザモードへ移行し，スタック上に積まれた座標値をサブルーチンgcopyに渡せる形に再構成してからgcopyをコールし，そののち再びB_SUPERでユーザーモードに戻している。

スーパーバイザモードへの移行にはいつものDOSコールsuperではなく，IOCSコールのB_SUPERを使ってみた。特に機能的な違いはない。ただ，superではスタックを介して引数の受け渡しを行うのに対し，B_SUPERではスタックの代わりにa1レジスタが使われる。なお，B_SUPERはすでにスーパーバイザモードなのにさらにスーパーバイザモードにしようとした場合にはd0に－1を返すことになっているので，34～35行でそのチェックをしている。関数が呼び出された時点ですでにスーパーバイザモードであれば，そのまま呼び出し元に戻るわけだ。

引数の再構成に関しては，特に説明するまでもないだろう。スタック上にロングワード単位で積まれた6つの値を，ワード単位でもう一度スタックに積み直している。その時点で，スタック上にサブルーチンgcopyの引数受け渡し領域と同じ構造のデータが出来上がるから，その先頭アドレス（＝sp）をgcopyに引き渡せばよい。

と，こうして作成した関数gcopy( )を使うときの注意点を挙げておこう。まず，当然，必要なモジュールをすべてリンクしてやらなければならない。_GCOPY.O（リスト4をアセンブルしたもの）とGCOPY.O（先月のリスト10をアセンブルしたもの），さらにはGCOPY.S内で外部参照されているいくつかのサブルーチン，データも忘れずにリンクする。実用上は，これら（＋なんならここ4回分の全サブルーチン）をひとまとめのライブラリにしておくのが便利だろう。

また，gcopy( )内ではグラフィック画面が初期化されているかどうか，画面モードがどうなっているのかのチェックは一切行っていない。画面の初期化は呼び出し元のCプログラム側で管理すること。

さらに，描画ページの切り替えの問題もある。gcopyは描画ページを自前で管理しており，XCのapage( )の影響を受けない。この問題は，リスト5のような関数（サブルーチン）を元のapage( )の代わりに使うことで回避できる。

ああ，まだあった。gcopyはアセンブル時に実画面1024×1024ドットモード用か，512×512ドットモード用かを決定するように作られていたのだった。1024ドットモード用と，512ドットモード用の2つのライブラリを作っておき，プログラムで使う画面モードに応じてリンクするライブラリを変えるなどの方法で逃げてもらいたい。

●

“Cの関数をマシン語で書く方法”だか，“最近作ったサブルーチンをCから使えるようにつじつま合わせする方法”だかわからなくなったところで，この話題は終わりとする。gcopy以外のほかのサブルーチンもまったく同じ方法でCの関数にできるから，

リスト5　_APAGE.S

```
 1:          .include        iocscall.mac
 2: *
 3:          .xdef   _apage
 4:          .xref   apage
 5: *
 6:          .text
 7:          .even
 8: *
 9: _apage:
10: arg =    4+3
11:          move.b  arg(sp),d1
12:          bmi     getapage
13:          ext.w   d1
14:          jmp     apage   *以前作ったサブルーチン
15:                          *引数はd1.wにページ番号
16: *
17: getapage:
18:          IOCS    _APAGE
19:          rts
20:
21:          .end
```

リスト6　GRAPH2.H

```
 1: /*  graph2.h    */
 2:
 3: #if !defined( __GRAPH2 )
 4:     #define __GRAPH2
 5:
 6:     #define _INT4   int, int, int, int
 7:     #define _INT5   int, int, int, int, int
 8:     #define _INT6   int, int, int, int, int, int
 9:     #define _INT4A  int, int, int, int, void *
10:     #define _INT4AI int, int, int, int, void *, int
11:
12:     void gfill( _INT5 );
13:     void gfill_or( _INT5 );
14:     void gfill_and( _INT5 );
15:     void gfill_xor( _INT5 );
16:     void gtilefill( _INT6 );
17:     void gnegate( _INT4 );
18:     void gmonotone( _INT4 );
19:     void gmonotone_y( _INT4 );
20:     void gsoftfocus( _INT4 );
21:     void goutline( _INT4 );
22:     void ghalftone( _INT4 );
23:     void gmonotone_hsv( _INT6 );
24:     void gmonotone_hsy( _INT6 );
25:     void gaddcolor( _INT5 );
26:     void gsubcolor( _INT5 );
27:     void gget( _INT4A );
28:     void gput( _INT4A );
29:     void gputon( _INT4AI );
30:     void gputin( _INT4AI );
31:     void ghalftoneput( _INT4A );
32:     void gcopy( _INT6 );
33:     void ghreverse( _INT4 );
34:     void gvreverse( _INT4 );
35:     void ghdup( _INT5 );
36:     void gvdup( _INT5 );
37:
38:     #undef  _INT4
39:     #undef  _INT5
40:     #undef  _INT6
41:     #undef  _INT4A
42:     #undef  _INT4AI
43:
44: #endif
```

各自やってみてもらいたい。おまけとして，リスト6に“そうやって作られるであろう”グラフィックライブラリ用のプロトタイプ宣言を含むヘッダファイルを示しておく。

で，次はX-BASICだ。

X-BASICの外部関数

X-BASICの外部関数の作成方法については『プログラマーズマニュアル』に詳しいし，本誌にもときおり思い出したように解説記事が載る。あえて補足することはなにもないのだが，とりあえず実例をひとつ示しておきたい。

半ばやけくそ気味で，最近作ったグラフィック関連サブルーチンを全部まとめたGRAPH2.FNCを作ってみる。ただし，この外部関数は画面モードが512×512ドット65536色モードでのみ使用可能とする。

X-BASICの外部関数ファイルはXファイルと同じ形式をとる。作成手順もふつうのマシン語プログラムと変わらない。ただし，プログラムの頭の部分にインフォメーションテーブルと呼ばれるヘッダをつけることになっている（図1）。実例を見ないとピンとこないかもしれないが，簡単に解説しておこう。

インフォメーションテーブルの頭の部分にはなにか特別なイベントが発生したときに呼び出される6つのルーチンの先頭アドレスが並ぶ。これらのすべてをサポートする必要はないが，サポートしないものについてはrts命令が置かれたアドレスを格納しておかなければならない[13]。続く2ロングワードは将来の拡張用で，いまのところはrtsが置かれたアドレスを入れておけばよい。

その後ろに，トークンテーブル，パラメータテーブル，関数の実行アドレステーブルの各先頭アドレスが続く。トークンテーブルは外部関数ファイル内の全関数名を0で終わる文字列の形で並べたものだ。テーブルの最後を示す意味でトークンテーブルの末尾にはもう1個余分な0を置く。

パラメータテーブルにはポインタが並び，“そのポインタの指す先”に，パラメータIDとか呼ばれているコードで各関数の引数/戻り値の型情報を置く。パラメータIDはひとつの引数/戻り値あたり16ビットで，各ビットには一応図2に示すような意味が割り振られている[14]。なお，XCにはFDEF.Hというファイルがついてきて，よく使われるパラメータIDがシンボル定義されている。

実行アドレステーブルは言葉どおり各関数の処理ルーチン本体の先頭アドレスを並べたものだ。

ここで，トークンテーブル，パラメータテーブル，実行アドレステーブル内の関数の並びはすべて同じ順序でなければならない。

さて，X-BASICの外部関数への引数受け渡しにはやはりスタックが用いられ，C同様，引数の並びの後ろから順に積まれる。ただし，スタックには引数の値に加えて，各引数の型，引数の総個数が積まれる（図3）。ひとつの引数あたり10バイト，うち，値の格納用に8バイトが割り当てられているわけだが，実際に8バイトすべてが使われるのはfloat型の場合だけで，int型の場合は下位4バイトのみ，char型の場合は下位の1バイトのみが使われ，残り

13) 外部関数がそのイベントをサポートしているかどうかに関わらず，X-BASICインタプリタはイベント発生時に無条件にこのテーブルを参照し，該当アドレスをサブルーチンコールする。

14) ただし，X-BASICはすべての組み合わせをサポートしているわけではないようだ。

図1　X-BASIC外部関数インフォメーションテーブル

+00$_{H}$	1L	X-BASIC起動時に呼び出されるサブルーチン先頭アドレス
+04$_{H}$	1L	RUN命令実行時に呼び出されるサブルーチン先頭アドレス
+08$_{H}$	1L	END命令実行時に呼び出されるサブルーチン先頭アドレス
+0C$_{H}$	1L	SYSTEM命令，EXIT( )関数実行時に呼び出されるサブルーチン先頭アドレス
+10$_{H}$	1L	BREAK, ^Cによりプログラムが中断されたときに呼び出されるサブルーチン先頭アドレス
+14$_{H}$	1L	^D入力時に呼び出されるサブルーチン先頭アドレス
+18$_{H}$	1L	予備
+1C$_{H}$	1L	予備
+20$_{H}$	1L	トークンテーブル先頭アドレス
+24$_{H}$	1L	パラメータテーブル先頭アドレス
+28$_{H}$	1L	実行アドレステーブル先頭アドレス
+2C$_{H}$	5L	予備(0で埋めておく)
⋮		
+3F$_{H}$		

図2　X-BASIC外部関数パラメータID

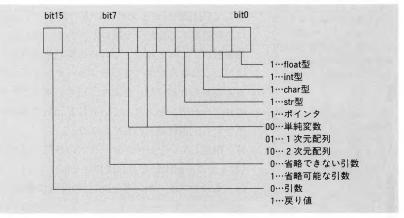



図3　X-BASIC外部関数へ渡される引数

SP+00$_{H}$	1L	リターンアドレス
SP+04$_{H}$	1W	引数の総個数
SP+06$_{H}$	1W	第1引数の型
SP+08$_{H}$	1L	第1引数の値(上位4バイト)
SP+0C$_{H}$	1L	第1引数の値(下位4バイト)
SP+10$_{H}$	1W	第2引数の型
SP+12$_{H}$	1L	第2引数の値(上位4バイト)
SP+16$_{H}$	1L	第2引数の値(下位4バイト)
SP+1A$_{H}$	1W	第3引数の型

図4　配列ポインタの構造(1次元配列の場合)

+00$_{H}$	1L	X-BASIC内部で使用
+04$_{H}$	1W	配列の次元数－1
+06$_{H}$	1W	配列要素1個のバイト数
+08$_{H}$	1W	配列の要素数－1(添え字の最大値)
+0A$_{H}$		配列の内容

は0で埋められる。また，str型/配列の場合は下位4バイトにデータへのポインタが格納されている。このポインタはstr型の場合は文字列の先頭を指すが，配列の場合は図4に示すような構造を持ったメモリ領域を指す。なお，図4は1次元配列の場合で，多次元配列の場合はもう少し情報量が増える。

関数からの戻り値は引数と同じような構造の10バイトの領域（ただし，型の部分は0にする）に値を収め，その先頭アドレスをa0に入れて返す。加えて，エラーなく処理が終了した場合にはd0.lに0を，エラーの場合は0以外のエラーコードを，a1にエラーメッセージへのポインタを返すことになっている。ただし，負のエラーコードにはHuman68kのDOSコールと同等の意味が割り当てられているようで，特にa1で指定しなくても適当なメッセージが表示されるようだ。外部関数特有のエラーメッセージを表示したい場合は，正のエラーコードを使えばよい。なお，正常終了時はa1は無視され，エラー時にはa0が無視される[15]。

では，リスト7を見てもらおう。各関数の実行ルーチン本体は含まれていない。引数や戻り値のつじつま合わせだけを行い，処理自体はすでに作ってあるサブルーチンにまかせるようになっている。

まず，インクルードファイルを取り込んでいる。FDEF.HはXCのものを流用しているが，もし不幸にしてFDEF.Hが手元にない人のために7～9行に注釈の形で必要部分を抜粋してある。続いて，外部参照がずらっと並ぶ。この外部関数を作成するには，ここに並んだすべてのサブルーチンをリンクする必要があるわけだ。たぶん，読者の多くはすべてのサブルーチンを入力しているほど元気じゃないと思うから，逃げ道を示しておこう。

1)　用意していないサブルーチンの外部参照を削る（たとえば，gfillがなければ13行を削除する）

2)　そのサブルーチン名を387行の直前に挿入して，rtsだけのサブルーチンにしてしまう（gfillの例だと，

　　gfill：

の1行を追加することになる）

スタックフレームなどのデータ構造のオフセット表定義に続いて，76行からがインフォメーションテーブルだ。イベントに対応した処理ルーチンは使う必要がなかったので，どれもrtsを指している。90行からのトークンテーブルは例の偶数境界の問題を避ける意味で最後に.even疑似命令が置いてある[16]。

その後ろ，121行以降がパラメータテーブル。先に触れたとおり，パラメータテーブルには“関数の引数/戻り値の個数/型を表す情報へのポインタ”を並べる。この外部関数ファイル中では引数の数や型が同一の関数が多いので，パラメータID列もある程度共通化してメモリを節約している。ちなみに，どの関数も戻り値はintで，正常終了時は0，エラー時は1を関数の戻り値とする（と決めた）[17]。

176行からの実行アドレステーブルはいいとして，207行から始まる各関数の処理ルーチンを見てもらおう。関数の多くは“引数の数”と“どの既存サブルーチンを呼び出すか”が異なるだけなので，処理の多くの部分は（紙面の無駄遣いを避ける意味で）やはり共通化してあり，なんだかんだでほとんどの関数は367行で合流している（その分braだらけになってしまったが）。

368～377行がエラーチェックを行っている部分で，IOCSコールAPAGEでグラフィック画面が初期化されているかどうかと，IOCSコールCRTMODにより画面モードが65536色モードかどうかを確認し，違っていたらその場でエラー処理ルーチンに飛んでエラー終了する。この時点でスタックにはいくつもの値が積みっ放しになっているが，396行のunlkでご破算になるから大丈夫。

ちょっと戻って，重要どころの配列の扱いを見てもらおう。329行以下の部分だ。gget( )やgput( )などの関数はパターンの格納領域として，任意の数値型の1次元配列を受け取る。このとき，その配列は，指定されたG-RAM上の領域を十分格納できるだけの大きさを持っていなければならない。配列の大きさは引数情報中の要素ひとつあたりのバイト数×要素数で計算できる。また，G-RAM上の領域が何バイト分かは縦横のドット数の積×2で得られる。比較の結果配列のバイト数のほうが小さければエラーだ[18]。

あと，パレットをまとめてセーブ/ロードする関数gsavepalet( )，gloadpalet( )は引数としてファイル名をとる形にした（292～314行）。元のサブルーチンはファイルハンドルを引数とするようになっていたが，X-BASICのファイル番号はファイルハンドルとは似てはいるが異なるものなので，外部関数の中でファイルのオープンからクローズまでをまとめてやってしまっている。期せずして，str型の引数をとる関数の単純な例になった。

●

以上，今回は高級言語とマシン語をリンクする方法を見てきた。この連載でこんなことをいってはまずいような気もするが，速度が要求される部分だけをマシン語で書いてメイン部分は高級言語で書くという手は，開発効率の点ではなかなかおいしい。逆に，マシン語の知識を持つことで高級言語の使い道にも幅が出てくる，といういい方もできるかもしれない。と強引に締めたところで，また来月。

15)　error offによりエラーを無視する設定になっているときには，エラー時にもa0は意味を持つ。

16)　実際に文字列が何文字かは数えていない。もし，もともとトークンテーブルが偶数バイトだった場合には.evenは単に無視されるだけだからだ。

17)　ここでいっている関数の戻り値はd0に返すエラーコードではなく，X-BASICプログラム側から見た関数の戻り値だが，結局はd0にも同じ値を返すように作ってある。

18)　gget( )はちょっと変な仕様にしてしまった関係で本来なら別扱いにするべきなのだが，手を抜いてgput( )などと処理を共有している。

```
  1:         .include       doscall.mac
  2:         .include       iocscall.mac
  3:         .include       fdef.h
  4:         .include       const.h
  5:         .include       gmacro.h
  6: *
  7: *int_val          equ      $0002   *int型の引数
  8: *str_val          equ      $0008   *str型の引数
  9: *aryl_fic         equ      $0037   *数値型1次元配列
 10: *int_ret          equ      $8001   *int型の戻り値
 11: IOCS_GL3          equ      12
 12: *
 13:         .xref   gfill
 14:         .xref   gfill_or
 15:         .xref   gfill_and
 16:         .xref   gfill_xor
 17:         .xref   gtilefill
 18:
 19:         .xref   gsavepalet
 20:         .xref   gloadpalet
 21:         .xref   gnegate
 22:         .xref   gmonotone
 23:         .xref   gmonotone_y
 24:         .xref   gsoftfocus
 25:         .xref   goutline
 26:         .xref   ghalftone
 27:         .xref   gmonotone_hsv
 28:         .xref   gmonotone_hsy     *
 29:         .xref   gaddcolor
 30:         .xref   gsubcolor
 31:
 32:         .xref   gget
 33:         .xref   gput
 34:         .xref   gputon
 35:         .xref   gputin
 36:         .xref   ghalftoneput
 37:         .xref   gcopy
 38:         .xref   ghreverse
 39:         .xref   gvreverse
 40:         .xref   ghdup
 41:         .xref   gvdup
 42: *
 43:         .offset 0        *スタックフレーム
 44: *
 45: A6BUF:  .ds.l   1
 46: RETADR: .ds.l   1
 47: PARC:   .ds.w   1
 48: PAR1:   .ds.b   10
 49: PAR2:   .ds.b   10
 50: PAR3:   .ds.b   10
 51: PAR4:   .ds.b   10
 52: PAR5:   .ds.b   10
 53: PAR6:   .ds.b   10
 54: PAR7:   .ds.b   10
 55: *
 56:         .offset 0        *パラメータバッファ
 57: *
 58: TYPE:   .ds.w   1        *型
 59: FVAL:   .ds.l   1        *実数
 60: PVAL:
 61: LVAL:   .ds.w   1        *32ビット数，ポインタ
 62: WVAL:   .ds.b   1        *16ビット数
 63: BVAL:   .ds.b   1        *8ビット数
 64: *
 65:         .offset 0        *X-BASICの配列
 66: *
 67: ASKIP:  .ds.w   1
 68: ADIM:   .ds.w   1        *次元数-1
 69: ASIZ:   .ds.w   1        *1要素のバイト数
 70: ALEN:   .ds.w   1        *要素数-1
 71: ADAT:                    *データ本体
 72: *
 73:         .text
 74:         .even
 75: *
 76: information_table:
 77:         .dc.l   dummy          *X-BASIC起動時
 78:         .dc.l   dummy          *run
 79:         .dc.l   dummy          *end
 80:         .dc.l   dummy          *system,exit
 81:         .dc.l   dummy          *BREAK,^C
 82:         .dc.l   dummy          *^D
 83:         .dc.l   dummy          *予備
 84:         .dc.l   dummy          *予備
 85:         .dc.l   token_table
 86:         .dc.l   param_table
 87:         .dc.l   exec_table
 88:         .dc.l   0,0,0,0,0      *予備
 89: *
 90: token_table:
 91:         .dc.b   'gfill',0
 92:         .dc.b   'gfill_or',0
 93:         .dc.b   'gfill_and',0
 94:         .dc.b   'gfill_xor',0
 95:         .dc.b   'gtilefill',0
 96:         .dc.b   'gsavepalet',0
 97:         .dc.b   'gloadpalet',0
 98:         .dc.b   'gnegate',0
 99:         .dc.b   'gmonotone',0
100:         .dc.b   'gmonotone_y',0
101:         .dc.b   'gsoftfocus',0
102:         .dc.b   'goutline',0
103:         .dc.b   'ghalftone',0
104:         .dc.b   'gmonotone_hsv',0
105:         .dc.b   'gmonotone_hsy',0
106:         .dc.b   'gaddcolor',0
107:         .dc.b   'gsubcolor',0
108:         .dc.b   'gget',0
109:         .dc.b   'gput',0
110:         .dc.b   'gputon',0
111:         .dc.b   'gputin',0
112:         .dc.b   'ghalftoneput',0
113:         .dc.b   'gcopy',0
114:         .dc.b   'ghreverse',0
115:         .dc.b   'gvreverse',0
116:         .dc.b   'ghdup',0
117:         .dc.b   'gvdup',0
118:         .dc.b   0              *テーブル終端
119:         .even
120: *
121: param_table:
122:         .dc.l   param_5i       *gfill
123:         .dc.l   param_5i       *gfill_or
124:         .dc.l   param_5i       *gfill_and
125:         .dc.l   param_5i       *gfill_xor
126:         .dc.l   param_6i       *gtilefill
127:         .dc.l   param_str      *gsavepalet
128:         .dc.l   param_str      *gloadpalet
129:         .dc.l   param_4i       *gnegate
130:         .dc.l   param_4i       *gmonotone
131:         .dc.l   param_4i       *gmonotone_y
132:         .dc.l   param_4i       *gsoftfocus
133:         .dc.l   param_4i       *goutline
134:         .dc.l   param_4i       *ghalftone
135:         .dc.l   param_6i       *gmonotone_hsv
136:         .dc.l   param_6i       *gmonotone_hsy
137:         .dc.l   param_5i       *gaddcolor
138:         .dc.l   param_5i       *gsubcolor
139:         .dc.l   param_4ia      *gget
140:         .dc.l   param_4ia      *gput
141:         .dc.l   param_4iai     *gputon
142:         .dc.l   param_4iai     *gputin
143:         .dc.l   param_4ia      *ghalftoneput
144:         .dc.l   param_6i       *gcopy
145:         .dc.l   param_4i       *ghreverse
146:         .dc.l   param_4i       *gvreverse
147:         .dc.l   param_5i       *ghdup
148:         .dc.l   param_5i       *gvdup
149: *
150: param_6i:       .dc.w   int_val *引数はint6個
151: param_5i:       .dc.w   int_val *引数はint5個
152: param_4i:       .dc.w   int_val *引数はint4個
153:                 .dc.w   int_val
154:                 .dc.w   int_val
155:                 .dc.w   int_val
156:                 .dc.w   int_ret *戻り値はint
157:
158: param_4ia:      .dc.w   int_val *引数はint4個+配列
159:                 .dc.w   int_val
160:                 .dc.w   int_val
161:                 .dc.w   int_val
162:                 .dc.w   aryl_fic
163:                 .dc.w   int_ret *戻り値はint
164:
165: param_4iai:     .dc.w   int_val *引数はint4個+配列+int
166:                 .dc.w   int_val
167:                 .dc.w   int_val
168:                 .dc.w   int_val
169:                 .dc.w   aryl_fic
170:                 .dc.w   int_val
171:                 .dc.w   int_ret *戻り値はint
172:
173: param_str:      .dc.w   str_val *引数はstr1個
174:                 .dc.w   int_ret *戻り値はint
175: *
176: exec_table:
177:         .dc.l   x_gfill
178:         .dc.l   x_gfill_or
179:         .dc.l   x_gfill_and
180:         .dc.l   x_gfill_xor
181:         .dc.l   x_gtilefill
182:         .dc.l   x_gsavepalet
183:         .dc.l   x_gloadpalet
184:         .dc.l   x_gnegate
185:         .dc.l   x_gmonotone
186:         .dc.l   x_gmonotone_y
187:         .dc.l   x_gsoftfocus
188:         .dc.l   x_goutline
189:         .dc.l   x_ghalftone
190:         .dc.l   x_gmonotone_hsv
191:         .dc.l   x_gmonotone_hsy
192:         .dc.l   x_gaddcolor
193:         .dc.l   x_gsubcolor
194:         .dc.l   x_gget
195:         .dc.l   x_gput
196:         .dc.l   x_gputon
197:         .dc.l   x_gputin
198:         .dc.l   x_ghalftoneput
199:         .dc.l   x_gcopy
200:         .dc.l   x_ghreverse
201:         .dc.l   x_gvreverse
202:         .dc.l   x_ghdup
203:         .dc.l   x_gvdup
204: *
```

```
205: *          引数がint 4 個の関数
206: *
207: x_gnegate:       lea.l   gnegate,a5
208:                  bra     exec_4i
209: *
210: x_gmonotone:     lea.l   gmonotone,a5
211:                  bra     exec_4i
212: *
213: x_gmonotone_y::  lea.l   gmonotone_y,a5
214:                  bra     exec_4i
215: *
216: x_gsoftfocus:    lea.l   gsoftfocus,a5
217:                  bra     exec_4i
218: *
219: x_goutline:      lea.l   goutline,a5
220:                  bra     exec_4i
221: *
222: x_ghalftone:     lea.l   ghalftone,a5
223:                  bra     exec_4i
224: *
225: x_ghreverse:     lea.l   ghreverse,a5
226:                  bra     exec_4i
227: *
228: x_gvreverse:     lea.l   gvreverse,a5
229:                  bra     exec_4i
230: *
231: *          引数がint 5 個の関数
232: *
233: x_gfill:         lea.l   gfill,a5
234:                  bra     exec_5i
235: *
236: x_gfill_and:     lea.l   gfill_and,a5
237:                  bra     exec_5i
238: *
239: x_gfill_or:      lea.l   gfill_or,a5
240:                  bra     exec_5i
241: *
242: x_gfill_xor:     lea.l   gfill_xor,a5
243:                  bra     exec_5i
244: *
245: x_gaddcolor:     lea.l   gaddcolor,a5
246:                  bra     exec_5i
247: *
248: x_gsubcolor:     lea.l   gsubcolor,a5
249:                  bra     exec_5i
250: *
251: x_ghdup:         lea.l   ghdup,a5
252:                  bra     exec_5i
253: *
254: x_gvdup:         lea.l   gvdup,a5
255:                  bra     exec_5i
256: *
257: *          引数がint 6 個の関数
258: *
259: x_gtilefill:     lea.l   gtilefill,a5
260:                  bra     exec_6i
261: *
262: x_gmonotone_hsv:lea.l    gmonotone_hsv,a5
263:                  bra     exec_6i
264: *
265: x_gmonotone_hsy:lea.l    gmonotone_hsy,a5
266:                  bra     exec_6i
267: *
268: x_gcopy:         lea.l   gcopy,a5
269:                  bra     exec_6i
270: *
271: *          引数がint 4 個+1次元配列の関数
272: *
273: x_gget:          lea.l   gget,a5
274:                  bra     exec_4ia
275: *
276: x_gput:          lea.l   gput,a5
277:                  bra     exec_4ia
278: *
279: x_ghalftoneput:  lea.l   ghalftoneput,a5
280:                  bra     exec_4ia
281: *
282: *          引数がint 4 個+1次元配列の関数+int 1 個の関数
283: *
284: x_gputon:        lea.l   gputon,a5
285:                  bra     exec_4iai
286: *
287: x_gputin:        lea.l   gputin,a5
288:                  bra     exec_4iai
289: *
290: *          パレットのロード/セーブ
291: *
292: x_gsavepalet:
293:         link    a6,#0
294:         move.w  #ARCHIVE,-(sp)
295:         move.l  PAR1+LVAL(a6),-(sp)
296:         DOS     _CREATE
297:
298:         move.w  d0,-(sp)
299:         bmi     done
300:         jsr     gsavepalet
301:         DOS     _CLOSE
302:
303:         bra     okret
304: *
305: x_gloadpalet:
306:         move.w  #ROPEN,-(sp)
307:         move.l  PAR1+LVAL(a6),-(sp)
308:         DOS     _OPEN
309:
310:         move.w  d0,-(sp)
311:         bmi     done
312:         jsr     gloadpalet
313:         DOS     _CLOSE
314:         bra     okret
315: *
316: exec_4iai:
317:         link    a6,#0
318:         move.w  PAR6+WVAL(a6),-(sp)
319:         bra     exec4a
320: *
321: exec_4ia:
322:         link    a6,#0
323: exec4a: movea.l PAR5+LVAL(a6),a0
324:         move.w  PAR4+WVAL(a6),d3
325:         move.w  PAR3+WVAL(a6),d2
326:         move.w  PAR2+WVAL(a6),d1
327:         move.w  PAR1+WVAL(a6),d0
328:
329:         pea.l   ADAT(a0)
330:         movem.w d0-d3,-(sp)
331:
332:         MINMAX  d2,d0           *x0<x1を保証
333:         MINMAX  d3,d1           *y0<y1を保証
334:         addq.w  #1,d1           *
335:         addq.w  #1,d3           *
336:         sub.w   d0,d2           *d2.w = x1-x0+1
337:         sub.w   d1,d3           *d3.w = y1-y0+1
338:         mulu.w  d3,d2           *d2.l = 領域のドット数
339:         add.l   d2,d2           *d2.l = 領域のバイト数
340:
341:         move.w  ALEN(a0),d0     *d0.w = 配列の要素数-1
342:         addq.w  #1,d0           *d0.w = 配列の要素数
343:         mulu.w  ASIZ(a0),d0     *d0.l = 配列のバイト数
344:
345:         cmp.l   d2,d0           *配列のほうが小さければ
346:         bcs     error           *  エラー
347:
348:         bra     exec
349: *
350: exec_6i:
351:         link    a6,#0
352:         move.w  PAR6+WVAL(a6),-(sp)
353:         bra     exec5
354: *
355: exec_5i:
356:         link    a6,#0
357: exec5:  move.w  PAR5+WVAL(a6),-(sp)
358:         bra     exec4
359: *
360: exec_4i:
361:         link    a6,#0
362: exec4:  move.w  PAR4+WVAL(a6),-(sp)
363:         move.w  PAR3+WVAL(a6),-(sp)
364:         move.w  PAR2+WVAL(a6),-(sp)
365:         move.w  PAR1+WVAL(a6),-(sp)
366:
367: exec:
368:         moveq.l #-1,d1          *グラフィック画面は
369:         IOCS    _APAGE          *
370:         tst.b   d0              *  初期化されているか?
371:         bmi     error
372:
373:         moveq.l #-1,d1          *画面モードは
374:         IOCS    _CRTMOD         *(d0=12,13...65536色モード)
375:         andi.b  #$fe,d0         *(第0ビットクリア)
376:         cmpi.b  #IOCS_GL3,d0    *  65536色モードか?
377:         bne     error
378:
379:         suba.l  a1,a1           *スーパーバイザモードへ
380:         IOCS    _B_SUPER        *
381:
382:         move.l  sp,-(sp)        *引数受け渡し
383:         jsr     (a5)            *実行ルーチン本体
384:                                 *d0.lは保存される
385:         tst.l   d0
386:         bmi     done
387:
388:         movea.l d0,a1           *ユーザーモードへ復帰
389:         IOCS    _B_SUPER        *
390:
391: okret:  moveq.l #0,d0           *正常終了
392:
393: done:   lea.l   retval,a0       *
394:         move.l  d0,LVAL(a0)     *戻り値
395:
396:         unlk    a6
397: dummy:  rts
398: *
399: error:
400:         lea.l   errmes(pc),a1   *エラーメッセージ
401:         moveq.l #1,d0           *エラーコード
402:         bra     done
403: *
404: errmes: .dc.b   'in GRAPH2.FNC',0
405:         .even
406: *
407:         .data
408:         .even
409: *
410: retval: .dc.w   0               *戻り値格納用
411:         .dc.l   0,0             *
412: *
413:         .end
```

●特集　XCのための傾向と対策

C Compiler

C compiler PRO-68K (XC) は，X68000のプログラミング環境を支える総合開発セットです。特に重要なのはX68000のハードウェアを隅々までサポートした豊富なライブラリ群で，話題のGNU　Cコンパイラを使用する場合にもこれらのライブラリがなくてはなりません。また，X-BASICのプログラムをコンパイルできるのも魅力です。そのためのライブラリも用意され，これは逆にC言語からBASICの関数が利用できるといったメリットもあるわけです。
そしてXCはバージョン2となり，プログラミングを支援する環境がさらに強化されました。その目玉がプログラムを実行しながらデバッグができるソースコードデバッガSCD.Xであり，また，モジュール化された大規模プログラム開発の際に役立つファイル保守ユーティリティMAKE.Xなどです。
今回はC言語そのものの魅力とともに，XCによってもたらされるプログラミングの世界を初心者の皆さんにもわかりやすく紹介していきましょう。

CONTENTS

Cコンパイラのアウトライン

XC ver.2.0ガイドマップ

Ogikubo Kei 荻窪　圭

スーパーヘビー級の開発セットC compiler PRO-68K ver.2.0。一口にXCといっても1本のツールではない。2枚組のシステムディスクの中には，それはもうたくさんのファイルが詰っているのだ。まずはこれらの中身を探索してみるとしよう。

あまりの重さに耐えかねたのか，組み上げると取っ手の出来上がるかさばる箱になったXC ver.2.0。最近あのタイプの箱は流行っているのか，DynaBook用パーソナルプリンタや，かのハンディ98も組み上げると取っ手のできる箱に入っていた。あれを買った人はみな重い思いをして持ち帰ったのだろう。ご苦労様。

*

無事帰宅して，取っ手に食い込んだ右手の指を剥がす。そして，2度3度右手を振って，感覚が戻るのを待っただろう。

やがて，自分の部屋へ重い箱を引きずっていき，わくわくしながら44,800円の箱を開ける。

箱の中に黒いボックスがあり，それごと，7冊のマニュアルと，黒いディスクケースを取り出す。箱はその辺にうっちゃっておき，マニュアルとディスクケースを眺める。

XC ver.1でばりばりやっていた貴方，きっと苦もなくver.2.0の世界へ行けたでしょう。けれど，初めてXCというものに触れた方，気の迷いで買ってしまった方。7冊のマニュアルのどれを見ても「導入マニュアル」だとか「はじめに読む本」なんてものはない。うーん。

答えは，「ユーザーズマニュアル」である。第1章に「お使いになる前に」とある。なかなかの不親切さ加減で，インストールについて書いてある。

XC ver.2.0のシステムディスクは2枚。けれど，そのままでは使えない。そこでインストールという作業が必要となる。必要なXCのファイル（パッケージ）を，使用するHuman68kのシステムに組み込んでやることだ。フロッピーベースだと実行用ディスクを1枚にまとめないと不便だからね。XC ver.2.0には，こういった実行用ディスクの作成や，ハードディスクにインストールするためのインストーラ“INSTALL.BAT”がついている。

私はインストーラを使わず，手作業でファイルを移した。理由はまあインストーラに対する不信感とか，添付のインストーラでインストールできるだけのハードディスクの空き容量がなかったことなどなどだ。

インストーラを使うにしても，手作業でコピーするにしても，必要なファイルとそうでないファイル，さらにはどのファイルはどんな役目を背負って生まれてきたかがわからないとドンキホーテだ。無謀だ。

初めてCコンパイラを手にする人は，そのファイルの多さにとまどう。はじめてのCやK&Rをどれだけ読んでも，そんな話は書いていない。プログラム開発前にこういったファイル構成や，各々の役割を知っておくと非常にあとが楽だ。

よって，以下の章は，XCワールドを歩きまわるためのガイドマップである。

システムディスク1を歩く

図1－1がシステムディスク1のディレクトリである。見てわかるとおり，ディレクトリは5つである。一見，Cに関係するファイルはないかの如く振る舞っているが，あにはからんや，そんなことはない。

XCとは何か。ただのCコンパイラではなく，X68000の開発環境セットなのである。

今のところXCについているのが最新バージョンだと思えるので，

“自分のX68000の環境を最新のものにリプレースする”

ことも考える必要がある。

特に，Human68kのバージョン1を使っていた人は，これを機にバージョン2にしてしまうのがよい。バージョン2の人も，今回のはver.2.02であるから，新しくしてしまおう。特に，SYSディレクトリにあるもの，BINディレクトリにあるものはいつの間にかバージョンアップしてたりするので，自分が使っているディスクに全部入れてしまおう。それが嫌な人は，ディレクトリを取り，日付の新しくなったものだけをコピーすればいい。COPYALLコマンドをうまく使えば，簡単にできる。なにげなく日付が新しくなっているものもある（図1－3）。そうでなくても，知らぬうちに自分のより新しいものがあったりするものだ。

さて，この中で何が必要であるか。

図1－2にシステムディスク1の全ファイルを示した。“おニュー”とあるのは，SX-WINDOW以降に追加されたものだ。今回のXCで初めてついたものもある。SUPER-HDについていたものもある。

“新ver.”とあるのは，とりあえず，なにげなくバージョンアップしていたものだ。日付が'90年のものにつけておいた。Human68kがSCSI対応になったことに対応するバージョンアップが多いと思われるが，そうでなくても，隠れて日付が変わってたりするので要注意だ。

“ver.確認”とあるのは，こういうファイルはまず疑ってみなさい，ってことだ。

1）SYSディレクトリ

あやしいのが“SCSIDRV.SYS”と“OPMDRV2.X”。前者がSCSIドライバってことはわかる。後者はあやしい。実はこれ，MIDI対応のOPMドライバなのだ。MIDIを持ってない人，持っていても間に合っている人は使う必要がないので，気にしなくてもいい。

気にすべきは，“IOCS.X”だな。これを使うと，テキスト表示やグラフィックが速くなるというものだ。使うも使わないも好き好きだが，私はCONFIG.SYSに組み込んで使っている。これを使うと，MZ-2500ばりのスムーズスクロールも体験できる。

2）BINディレクトリ

こいつもあやしい。“*”のついているのが，とりあえず開発必需品（あるいは準必需品）だ。このうち，なにがなんでも必要なのが“LK.X”，つまりリンカだな。こいつは新バージョンでないとコンパイルできない。あとのは，コンパイルするだけならなくてもいい。コンパイルするだけ，

ならね。

まあ，いろいろと便利なものが多いので，全部インストールしてしまおう。

3) その他

とりあえず，ETCディレクトリはいらない。インストールしてしまえば用済みだ。

システムディスク2を歩く

いよいよコンパイラ本体の登場だ。図2－1は図1－1と同様，システムディスク2の中のディレクトリである。これだけある。SAMPLE以外は必要そうなものばかりだ。

いきなり，図2－2の全ファイル図を覗いて，そのガイドを始めることにする。

1) CCディレクトリ

C Compilerってわけで，CCである。かつて，このディレクトリにはCC0とかCC1とかCCPとかいろんな子供がいたが，今回の新バージョンでは“CC.X”ひとつしかない。普通に考えて，“前のバージョンでいろいろ分かれていたプログラムを1

図1-1 システムディスク1のディレクトリ

```
¥
|
|-SYS
|
|-BIN
|
|-ASK
|
|-HIS
|
|-ETC
```

図2-1 システムディスクのディレクトリ

```
¥
|
|-CC
|
|-INCLUDE
|
|-LIB
|
|-BASIC2
|
|-BC
|
|-SAMPLE
```

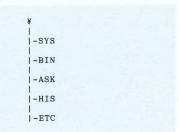
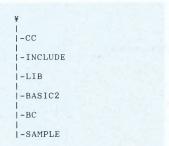



図1-3 SYSディレクトリの内容

XCシステム#1　　B:¥SYS
14 ファイル　　972K Byte 使用中　　249K Byte 使用可能
ファイル使用量　　238K Byte 使用

PRNDRV	SYS	1816	89-02-10	12:00:00	
PCMDRV	SYS	416	87-05-15	12:00:00	
ASK68K	SYS	121470	89-04-04	12:00:00	
RAMDISK	SYS	1816	89-02-10	12:00:00	
SRAMDISK	SYS	924	87-05-15	12:00:00	
PRNDRV1	SYS	3566	87-05-15	12:00:00	
PRNDRV2	SYS	1816	87-05-15	12:00:00	
PRNDRV3	SYS	1816	87-05-15	12:00:00	
SCSIDRV	SYS	1274	90-06-15	12:00:00	
OPMDRV2	X	35798	90-05-05	12:00:00	
FLOAT1	X	11498	87-11-03	12:00:00	
FLOAT2	X	12844	90-05-05	12:00:00	←いつの間にか新しくなった
HISTORY	X	27830	89-04-04	12:00:00	
IOCS	X	14420	90-06-15	12:00:00	

図1-2 システムディスク1の全ファイル

```
¥
|-XCシステム#1          ←ディスクの名前
|-HUMAN.SYS             ←Ver.確認
|-CONFIG.SYS
|-KEY.SYS
|-USKCG.SYS
|-BEEP.SYS
|-STARTUP.ENV
|-AUTOEXEC.BAT          ←環境変数をチェック
|-COMMAND.X           ←Ver.確認
|
|-SYS
|     |-PRNDRV.SYS
|     |-PCMDRV.SYS
|     |-ASK68K.SYS
|     |-RAMDISK.SYS
|     |-SRAMDISK.SYS
|     |-PRNDRV1.SYS
|     |-PRNDRV2.SYS
|     |-PRNDRV3.SYS
|     |-SCSIDRV.SYS     ←おニュー
|     |-OPMDRV2.X       ←おニュー
|     |-FLOAT1.X
|     |-FLOAT2.X        ←新Ver.
|     |-HISTORY.X
|     |-IOCS.X          ←おニュー
|
|-BIN
|    *|-AS.X            ←新Ver.
|    *|-LK.X            ←新Ver.
|    *|-DB.X            ←新Ver.
|    *|-SCD.X           ←おニュー
|    *|-SCD.CNF         ←おニュー
|    *|-SCD.HLP         ←おニュー
|     |-AR.X
|    *|-LIB.X           ←おニュー
|    *|-CV.X            ←新Ver.
|     |-BIND.X
|     |-CASE.X
|     |-DRIVE.X         ←新Ver.
|     |-MOVE.X
|     |-PROCESS.X
|     |-COPYALL.X       ←新Ver.
|     |-TERM.X
|     |-TOUCH.X         ←新Ver.
|     |-TREE.X
|     |-WHERE.X
|     |-FORMAT.X        ←新Ver.
|     |-DISKCOPY.X      ←新Ver.
|    *|-MAKE.X          ←おニュー
|     |-PRINT.X         ←新Ver.
|    *|-ED.X            ←新Ver.
|    *|-ED.HLP
|
|-ASK
|     |-ENV1.ASK
|     |-ENV2.ASK
|     |-ENV3.ASK
|     |-ENV4.ASK
|     |-ENV5.ASK
|
|-HIS
|     |-KEY.HIS
|     |-HISTORY.HIS
|     |-HISTORY.HLP
|
|-ETC
|     |-INSTALL.BAT     ←インストーラ
|     |-FD.BAT          ←インストーラで使用
|     |-FDR.BAT         ←〃
|     |-HD.BAT          ←〃
|     |-YN              ←〃
|     |-TOOL.X          ←〃
```

図2-2 システムディスク2の全ファイル

```
¥
|-XCシステム#2          ←ディスクの名前
|
|-CC
|     |-CC.X            ←Cコンパイラ
|
|-INCLUDE               ←インクルードファイル
|     |-ASSERT.H          が入っている
|     |-AUDIO.H
|     |-BASIC.H
|     |-BASIC0.H
|     |-CLASS.H
|     |-CONIO.H
|     |-CTYPE.H
|     |-DIRECT.H
|     |-DOSLIB.H
|     |-ERROR.H
|     |-FCNTL.H
|     |-FCTYPE.H
|     |-FLOAT.H
|     |-GRAPH.H
|     |-IMAGE.H
|     |-IO.H
|     |-IOCSLIB.H
|     |-JFCTYPE.H
|     |-JSTRING.H
|     |-LIMITS.H
|     |-MATH.H
|     |-MOUSE.H
|     |-MUSIC.H
|     |-MUSIC2.H
|     |-PROCESS.H
|     |-SETJMP.H
|     |-SIGNAL.H
|     |-SPRITE.H
|     |-STAT.H
|     |-STDARG.H
|     |-STDDEF.H
|     |-STDIO.H
|     |-STDLIB.H
|     |-STICK.H
|     |-STRING.H
|     |-TIME.H
|     |-TIMEB.H
|     |-UTIME.H
|     |-DOSCALL.MAC
|     |-ERROR.MAC
|     |-FCNTL.MAC
|     |-FDEF.H
|     |-FEFUNC.H
|     |-IOCSCALL.MAC
|     |-LIMITS.MAC
|     |-MALLOC.MAC
|     |-MATH.MAC
|     |-PROCESS.MAC
|     |-STAT.MAC
|     |-STDIO.MAC
|     |-TIME.MAC
|
|-LIB                   ←ライブラリ
|     |-CLIB.L
|     |-BASLIB.L
|     |-DOSLIB.L
|     |-IOCSLIB.L
|     |-FLOATFNC.L
|     |-FLOATEML.L
|
|-BASIC2                ←BASIC
|     |-BASIC.X         ←新しいVer.
|     |-BASIC.CNF
|     |-MUSIC2.FNC      ←おニュー
|     |-AUDIO.FNC
|     |-GRAPH.FNC
|     |-MOUSE.FNC
|     |-STICK.FNC
|     |-IMAGE.FNC
|     |-SPRITE.FNC
|
|-BC                    ←BASIC→Cコンバータ
|     |-BC.X
|     |-AUDIO.DEF
|     |-MOUSE.DEF
|     |-SPRITE.DEF
|     |-STICK.DEF
|     |-IMAGE.DEF
|     |-GRAPH.DEF
|     |-BASIC.DEF
|     |-MUSIC2.DEF      ←おニュー
|     |-BASIC.CNF
|
|-SAMPLE                ←サンプルプログラム
|     |-SAMPLE.DOC
|     |-MESH.C
|     |-DEF.BAS
|     |-DEF.HLP
|     |-MAKEFILE
```

本にしたな”って結論が導き出される。そのとおりで，CC.Xも400Kバイト以上と，破格の大きさだ。とっても大きい。おかげでメインメモリ1Mバイトユーザーは泣く泣く2Mバイトに増設するか，泣く泣く日本語FEPなどを削って最小限の環境で使うか，どっちにしても泣くしかなくなった。「Cコンパイラを買うくらいのユーザーは，きっと，メモリの増設くらいしてるだろう」って思ったのかもしれない。

CC.Xに統合化された機能は次のとおり。

CC.X 以下のプログラムをコントロールするメインプログラム。

BC.X BASICプログラムをCプログラムに変換するためのもの。

CCP.X PはプリプロセッサのP。プリプロセッサってのはプリプロセスを行うプログラムのことで，プリ（PRE）は“あらかじめ”でプロセスは“処理”だから，ここでは“コンパイルの前にあらかじめ行う処理をするプログラム”って意味だ。

後述のインクルードファイルのところで，こいつの役割を解説しよう。

CC0.X,CC1.X CCPが解析したプログラムをアセンブラのソースプログラムに変換する，コンパイラの本体部分。

CC2.X オプティマイザといわれる。CC1.Xの吐き出した出力を最適化したアセンブラのプログラムに変換するものだが，たいしたことはしてくれなかったらしい。

AS.X アセンブラである。アセンブラのソースプログラムをアセンブルするプログラム（ああ，わかんない人が読むと何がなんだかわかんない文章だ）である。AS.Xはシステムディスク1に入っていたではないか,ってな話もあるが，あれは，アセンブラで開発する人のためである。

と，この7本がひとつになったのだから，でかくて当たり前かもしれない。これによって，全体のコンパイルに要する時間は短くなったらしい。が，CC.Xを読む時間が長くなったので，えんえんとCC.Xを読んだはいいがエラーがたくさん出てあっさりと終わり，なんてときは腹が立つぞ。

2) INCLUDEディレクトリ

インクルードファイルが入っているディレクトリだ。includeというのは“(全体の一部として）含める”っていう意味。

includeディレクトリには拡張子が“.H”のファイル41個と，拡張子が“.MAC”のファイル11個が入っている。

Hってのは“ヘッダ”, MACってのは“マクロ”の略号である。まず，Cに馴染み深いヘッダファイルから見ていこう。

図3 MATH.Hの中身

```
/*
 * math.h X68k XC Compiler v2.00 Copyright 1990 SHARP/Hudson
 */
#ifdef  FORWORD
#define __PROTO_TYPE
#endif
#ifdef  __STDC__
#define __PROTO_TYPE
#endif

/***** math exceptions ****/
struct  exception       {
        int     type;
        char    *name;
        double  arg1, arg2;
        double  retval;
};

struct  complex {
        double  x,y;
};

/** Exception type codes **/
#define DOMAIN          1
#define SING            2
#define OVERFLOW        3
#define UNDERFLOW       4
#define TLOSS           5
#define PLOSS           6

#define PI              3.1415926535898
#define PID2            1.5707963267949         /* PI/2                 */
#define PID4            0.78539816339745        /* PI/4                 */
#define I_PI            0.31830988618379        /* 1/PI                 */
#define I_PID2          0.63661977236758        /* 1/PID2               */
#ifndef HUGE_VAL
#define HUGE_VAL        3.5953862697247E+308    /* huge double value    */
/*                      1.1125369292536E-308                            */
#endif

extern  volatile int    errno;

#ifdef  __PROTO_TYPE

double  acos(double);
double  asin(double);
double  atan(double);
double  atan2(double, double);
double  cos(double);
double  sin(double);
double  tan(double);
double  cosh(double);
double  sinh(double);
double  tanh(double);
double  exp(double);
double  frexp(double, int *);
double  ldexp(double, int);
double  log(double);
double  log10(double);
double  modf(double, double *);
double  pow(double, double);
double  sqrt(double);
double  ceil(double);
double  fabs(double);
double  floor(double);
double  fmod(double, double);
double  hypot(double, double);
double  except(int, char *, double, double, double);
int     matherr(struct exception *);
double  cabs(struct complex *);

#undef  __PROTO_TYPE
#else

double  acos();
double  asin();
double  atan();
double  atan2();
double  cos();
double  sin();
double  tan();
double  cosh();
double  sinh();
double  tanh();
double  exp();
double  frexp();
double  ldexp();
double  log();
double  log10();
double  modf();
double  pow();
double  sqrt();
double  ceil();
double  fabs();
double  floor();
double  fmod();
double  hypot();
double  except();
int     matherr();
double  cabs();

#endif
```

うんちゃら.HってファイルはすべてCのインクルードファイルである。

たとえば，以下のプログラムがあったとする。cos(π)の値を求めている例だ（ただし，このままではなんの役にも立たない。だって，結果をどこにも出力していないんだもん)。

例1）

```
#include    <MATH.H>
double      K;
main( )
{
        K=cos(PI);
}
```

この一番上の#includeってのがインクルードファイルを呼び出すプリプロセスコマンドである。頭に#がついているとプリプロセスコマンドだよ，っていう印になり，includeってのは，次のファイル名のファイルをコンパイルの前にこの位置にくっつけてちょ！　てな意味となる。こいつはプリプロセッサの仕事で，CC.Xの中に含まれている。

さて，では，このMATH.Hはどうなっているか。それが図3である。中身はテキストファイルであり，こうなっているのだ。中に#うんちゃらっていうコマンドが山ほどあるが，それは後ろで中森氏が説明してくれるはずだ。この中で，MATH，つまり数学関数で使う定数や関数の名前や構造の定義なんかをしているのがわかるはずだ。C自体は“PI”なんていう定数は持っていないのだが，ここで定義しているので，#includeしておけば，勝手にプログラム中で使ってかまわないのである。

さて，こいつがプリプロセスの段階でプログラムにくっつくわけであるが，実のところ，インクルードファイルの中の#うんちゃらもプリプロセスコマンドなのでプリプロセッサに解釈され，その結果だけがプログラムにくっつく。

具体例を見てみよう。

先の例1をプリプロセッサにかけた結果が図4だ。こいつはCC.Xの/Pオプションで見ることができる。#うんちゃらがなくなったのがわかってもらえるだろうか。数学的関数の名前がずらっと宣言されたのがわかるだろうか。

そんでもって，#defineってやつで図3で定義されていたPI（πの値）が実際の数字に置き換わったのがわかるだろうか。

プリプロセッサはこういう仕事をするのであり，プリプロセスコマンドはこう使うのである。

インクルードファイルはこのように単純なテキストファイルなので，自分で勝手に“OGIKUBO.H”なんてファイルを作って，中で勝手に，

```
#define  OGIKUBO     1024
```

なんてしてやると，プログラム中で，

```
K = OGIKUBO*2;
```

って書くだけで，Kに2048っていう値が求められるのだ。なんて具体的な説明だ。

ここでちょいと図4に戻る。中にある，

“#　数字　ファイル名”

の謎だ。この数字が何を表すコードかってのはさておいて，MATH.Hっていうファイル名がフルパスで書かれている。どうしてフルパスで書かれているのか。これはプリプロセッサがいろんなディレクトリを探してやっとこさMATH.Hを探し当てたっていう意味ではない。

あらかじめ，Human68kの環境変数ってやつに，

include=D:¥XC¥INCLUDE

ってのが定義してあったからだ。プリプロセッサはインクルードファイルを探すとき，includeっていう環境変数を参照するのだ。だから，includeって環境変数名に割り当てたディレクトリか，カレントディレクトリか，コンパイル時のオプションで指定したディレクトリにうんちゃら.Hを置いておく必要があるのだ。これが，図1－2の“環境変数をチェック”の意味だったりする。

INCLUDEディレクトリの話からプリプロセスの話まで流れてしまったが，まあ，避けて通れないこともないが，わかっているとなおおいしいっていう道なので，お得なのだ。

ちょっとだけ付け加えておく。MACファイルの話だ。こいつはアセンブラで開発する人以外はきっといらないので，気にしないようにしよう。アセンブラで使う汎用マクロファイルが入っているのだ。

3）LIBディレクトリ

CC.XはCで書かれたプログラムを最終的にアセンブルまでする。ver.1ではアセンブラのプログラム（“うんちゃら.S”ってやつ）を出力したけど，ver.2ではそれがない。直接，アセンブル後の“うんちゃら.O”というオブジェクトプログラムが出力される。

さて，INCLUDEのところで書いたように，#includeで読み込んだ“うんちゃら.H”ファイルには，関数の宣言が入っていた。というより，宣言しか入っていない。その関数の本体，つまり実際に処理をする部分はそこにあらず，なのである。さすがに宣言だけすればいいというわけにもいかない。そんなにこの世は甘くない。

図4　プリプロセッサのお仕事例

```
# 1 "fig7.c"

# 1 "D:¥XC¥include¥math.h"

# 6 "D:¥XC¥include¥math.h"


struct  exception       {
        int     type;
        char    *name;
        double  arg1, arg2;
        double  retval;
};

struct  complex {
        double  x,y;
};


extern  volatile int    errno;


double  acos(double);
double  asin(double);
double  atan(double);
double  atan2(double, double);
double  cos(double);
double  sin(double);
double  tan(double);
double  cosh(double);
double  sinh(double);
double  tanh(double);
double  exp(double);
double  frexp(double, int *);
double  ldexp(double, int);
double  log(double);
double  log10(double);
double  modf(double, double *);
double  pow(double, double);
double  sqrt(double);
double  ceil(double);
double  fabs(double);
double  floor(double);
double  fmod(double, double);
double  hypot(double, double);
double  except(int, char *, double, double, double);
int     matherr(struct exception *);
double  cabs(struct complex *);


# 102 "D:¥XC¥include¥math.h"

# 2 "fig7.c"

double  K;
main()
{
        K=cos(3.1415926535898);
}
```

どこに入っているか。

結果として，必要な関数の機能を記述したプログラムが“うんちゃら.X”になった状態でその中に組み込まれていれば問題なく動作するわけである。実際，問題なく組み込まれることになっている。

それが，LIBディレクトリに入っている“なんちゃら.L”というライブラリだ。LIBというのはライブラリ（LIBRARY）なわけだね。

ここにはインクルードディレクトリとは違って，6つしか入っていない。名前を見ればわかるとおり，C専用関数のライブラリとBASIC TO C用のBASICと同じ関数のライブラリと，DOSコールの機能を関数にしたやつのライブラリと，IOCSコールの機能を関数にしたやつのライブラリ，浮動小数点演算関係のライブラリだ。これらにはいろんな関数のオブジェクトプログラムがかき集められて，ぐしゃっと詰まっている。

CC.Xが吐き出したオブジェクトプログラムに，このライブラリから必要なものを組み込んでやらねばならないわけだ。

それがリンカ（LK.X）の仕事である。どの関数を使うよん，ってのはこっそりと“うんちゃら.O”の中に書いてあるので，リンカさんはライブラリから必要なものだけを“うんちゃら.O”に組み込んで，単体で実行できる形式にしてやるのだ。この作業をリンクという。

そして，LK.Xが“なんちゃら.X”を吐き出して，無事，コンパイルは終了するわけだ。あとは，“なんちゃら”って，キーボードから打ち込んで，うまく動くのを確認すれば一件落着である。

図1－2のBINディレクトリにあった“LIB.X”ってのはユーザーが自分で“なんちゃら.LIB”を作ったり，“なんちゃら.LIB”に新しく追加したりするツールだ。ライブラリアンという。

includeがそうだったように，ライブラリも“ライブラリはここにあるよん”と教えてやる必要がある。そのための環境変数がlibであり，

set lib＝D:￥XC￥LIB

のように，AUTOEXEC.BATかどこかで指定しておく必要がある。

4) BASIC2

BASIC2ディレクトリ。BASIC.X，つまりBASICインタプリタが入っているディレクトリである。BASIC.Xは前の“DRIVE.Xコマンドによる論理ドライブの変更や，SUBSTコマンドによる仮想ドライブに対応していなかった”ボケが改善されたなどの変更があるようだ（つまり，Human68k ver.2対応になった）。

MUSIC2.FNCはMIDI関係の関数が追加になった。OPMDRV2.Xを組み込んで，MIDIを使う人が使う。そうでなくて，OPMDとかを使う人は使わない。

5) BC

BCディレクトリはBASIC TO Cコンバータ関係のディレクトリである。BASICで書かれたプログラムをCに変換するときに使い，BC.XにBASICプログラムを放り込むと，Cのプログラムが生成される。

BC.Xとともにずらっと並んだ“.DEF”ファイルはBASIC関数の定義ファイル。DEFというのは常々，DEFINEの略語である。図5はMUSIC.DEFの中身。このように，BASICで使用する関数の定義が入っていて，どの関数群を使うかはBASICインタプリタと同様，“BASIC.CNF”に書いてある。CNFってのはCONFIGURATIONの略だ。

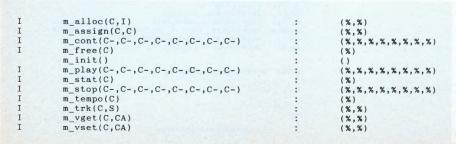
CC　うんちゃら.BAS

ってのを実行すると，CC.Xは“うんちゃら.C”，さらに“うんちゃら.O”を作成し，LK.Xのお世話になって，“うんちゃら.X”を作るのだ。

CC.Xさんが.DEFはどこかいな，って探すのにも環境変数を使う。

set bc＝D:￥XC￥BC

といった感じで，環境変数bcに設定したディレクトリがそうだ。ちゃんとセットしておくように。

6) SAMPLE

SAMPLE.DOCはまあ，このディレクトリの中身の使い方だな。MESH.Cはともかくとして，DEF.BASはBASICのサンプルでついてきたスプライトエディタだが，コンパイルすると速くなるので便利。

MAKEFILEってのは，MAKE.XっていうUNIXゆずりのMAKEコマンド用のサンプルだ。MAKE.Xは図1－2のBINのディレクトリに入っている。

図5　MUSIC.DEFの中身

```
I       m_alloc(C,I)                              :          (%,%)
I       m_assign(C,C)                             :          (%,%)
I       m_cont(C-,C-,C-,C-,C-,C-,C-,C-)           :          (%,%,%,%,%,%,%,%)
I       m_free(C)                                 :          (%)
        m_init()                                  :          ()
I       m_play(C-,C-,C-,C-,C-,C-,C-,C-)           :          (%,%,%,%,%,%,%,%)
I       m_stat(C)                                 :          (%)
I       m_stop(C-,C-,C-,C-,C-,C-,C-,C-)           :          (%,%,%,%,%,%,%,%)
I       m_tempo(C)                                :          (%)
I       m_trk(C,S)                                :          (%,%)
I       m_vget(C,CA)                              :          (%,%)
I       m_vset(C,CA)                              :          (%,%)
```

ライブラリディスクを横目で眺める

3枚目がいらない人はいらないというディスクである。これはなくともコンパイルはできる。図にするまでもなく，入っているファイルは2つだけだ。

CLIB.ARCとBASLIB.ARCである。

これは拡張子からわかるとおり，アーカイブファイルである。ARCはARCHIVESの略で，記録保管所という意味。いろんなファイルを1本にまとめつつも，中ではすぐ必要なファイルが取り出せるようにちゃんと並んでいるものを指す。

BASLIB.ARCをダンプしてみると（コントロールコードが入っているのでTYPEコマンドでは見られないが），BASIC関係のCの関数のアセンブラのソースが全部詰まっているのが見える。それを読んで勉強しても，必要なものだけ取り出してアセンブラプログラムを作るときに活用してもいい。

DOSLIB.ARCとIOCSLIB.ARCがないなどといわないように。あれらはそのままIOCSやDOSのルーチンをコールしているだけなので，アセンブラのソースをわざわざ載せることもあるまい。

XCへの道

これで，XCへの道は開かれたわけである。これからCに挑戦しようという皆様方，“べつにCでなくてもいいじゃん”てなことでもCを使ってやってください。

暇潰しプログラミング程度の短いやつでも，1回作ってしまえばいつでもコマンドシェルから呼び出せるっていうおいしさはBASICではなかなか味わえない。

あと，注意すべきは，“資源”ですな。ハードディスクは必需品。ハードディスクがなくてもRAMが4～6Mバイトくらい載っていれば大きなRAMディスクを確保できるからいいけどね。

というわけで，原の落球を見ながら（うふうふ）この原稿は終わる。

Cコンパイラが取り扱うファイル

ひとりのプログラマがCでプログラムを作ろうと思い立ってからそのプログラムが完成するまで，いろいろなファイルが作成され，参照され，そして生成される。順番に見ていこう。

まず，以下の説明でちょくちょく顔を出す「スイッチ」という用語は，コンパイルの条件を設定するためにプログラマがコマンドラインから指定する文字列で，コンパイルオプションとも呼ばれる。たとえば，

CC /W /Y myprog.c

としてコンパイルするとき，Cコンパイラに/Wおよび/Yというスイッチを与えたことになる。スイッチはスラッシュ（除算記号「/」やハイフン（減算記号「-」））のあとに英数字をくっつけた形で与える習慣になっている。XCはどちらも認識してくれるようだが，GCC（GNU C）はハイフンしかスイッチとして認識しない。

*

●〜.c（作成）

いわずと知れたCのソースファイル。基本的には，プログラマはこのファイルだけをED.Xなどのテキストエディタで書いてコンパイラにかければよい。

●〜.h（作成または参照）

ヘッダ（header）ファイルの略。プログラムの先頭には，

#include<stdio.h>
#include "myheader.h"

などを書いてヘッダファイルを指定するのが習慣になっているからである。

ヘッダの役割は，主に，ライブラリに格納されている関数の宣言，ライブラリ関数で用いる定数や構造体などの宣言，マクロの定義などである。通常ヘッダファイルには関数の本体（プログラム）は書かない。関数本体は，あらかじめコンパイルしてライブラリ化し，ユーザープログラムをコンパイルする際にリンクするほうが，コンパイル時間が短くなるからだ。

プリプロセッサ命令#includeのあとに，ヘッダファイル名を，

＜（ファイル名）＞

と不等号で囲むと，環境変数includeで指定されたパス（システム標準のヘッダファイルを格納しているディレクトリ）だけから指定されたヘッダファイルを探しにいく。

"（ファイル名）"

のようにダブルクォートで囲んだ場合，環境変数includeが指すディレクトリの前に，カレントディレクトリも調べることになっている。

たとえば，＜＞はシステムが付けてくれる標準のヘッダファイルを指定するのに用い，対して自前で作ったヘッダファイルはカレントディレクトリに置いて，" "で指定するといった使い分けをするときに都合がいい。

●〜.p（生成）

XCで/Pスイッチをつけたときに生成されるプリプロセッサの出力で，通常のコンパイルで表に現れることはない。プリプロセッサとは，ソースファイルがCコンパイラ本体で処理できるような形に加工するプログラム。具体的にはインクルードファイルやマクロの指定を展開する。コンパイルの前処理（プリプロセス）を行うという意味でプリプロセッサと呼ぶ。

GCCでは-Eスイッチでプリプロセッサの出力が標準出力（つまり画面）に出てくる。ファイルに保存したいときはリダイレクションを用いるとよい。

●〜.s（作成または生成）

多くのCコンパイラは，直接オブジェクトを出力することはしない。Cコンパイラの仕事は，パーサやコードジェネレータを呼び出し，ソースファイルを解釈してアセンブルリストを出力することであり，アセンブラを呼び出してオブジェクトファイルを生成させることである。コンパイラが出力するアセンブラのソースファイルがこの〜.sファイルだ。

ただし，GCCでは実行時に特にスイッチ(-S)をつけない限り，生成されない。アセンブラのソースは，実行に必要ないのだ。余分なファイルは要求されない限り残さないというのがGCCの作法であるように思われる。

●〜.mac（参照または作成）

アセンブルする段階で，IOCSコールやDOSコールを使うことは多い。その関連のマクロを集めたのが，ヘッダと同じく環境変数includeのディレクトリに入っている，

iocscall.mac
doscall.mac

である。なお，上で書いたように〜.sを自分で作るときなどは，このマクロ定義ファイルを自分で書く可能性もある。

●〜.o（生成）

アセンブラが出力するオブジェクトファイル。このままではまだ実行できない。最後のリンクフェイズを経て，ようやく一人前の実行可能ファイルになる。

GCCでは実行時に特にスイッチ(-Fc)をつけない限り，生成されない。このファイルも初めのうちは必要ないだろう。自分でライブラリを作るようになれば，〜.oファイルを作る必要も出てくる。

●〜.a(.l)（参照または作成）

Cプログラミングにおける影の立て役者，それがこのライブラリファイルである。

このライブラリファイルの正体は，たくさんのオブジェクトファイルである。Cで利用できる関数はこの中に収められている。リンカは，この中からユーザープログラムで参照されている関数（のオブジェクト）だけを切り出し，先ほどのオブジェクトプログラムと結合する。

Cのライブラリ関数はあまりにも数が多いので，それをいくつかにグループ分けし，ひとつのグループにつきひとつのファイルにまとめている。おかげで，ライブラリのディレクトリはすっきりしたものである。

バージョン1では，アーカイバというプログラムでまとめた〜.aというファイルだったが，バージョン2ではライブラリアンというプログラムでライブラリ化し，拡張子も〜.lである。したがって，旧バージョンのリンカでは，新しいライブラリを使った場合，素直にコンパイルできなくなっている。

インクルードファイルが格納されているのは環境変数includeが指すディレクトリであったが，ライブラリは環境変数libが指すディレクトリに格納されている。

XCでは，一部のライブラリ（clib.a）以外を用いるとき，/W, /Yスイッチが必要になっている。

ライブラリもユーザーが作ることができる。簡単に説明しておくと，まずCで関数を書く。このときmain()関数は書かない，つまり完結したプログラムにしない。それをコンパイルする。ただし実行ファイル（〜.x）は作らずに，オブジェクトファイル（〜.o）の段階で止めておく。スイッチはXCなら/Fc，GCCなら-c。それをアーカイバやライブラリアンで，ライブラリファイル，たとえばmyclib.aやmyclib.lにまとめる。

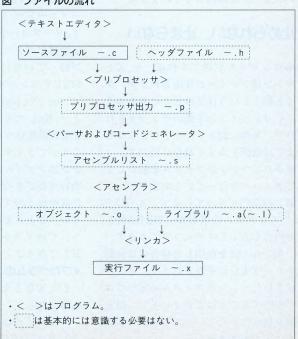
●〜.x（生成）

オブジェクトファイルとライブラリをリンカで結合して，ようやく出来上がった実行可能ファイルがこれ。もう，ふつうのDOS外部コマンドと同様に呼び出して使うことができる。

*

以上をまとめる。実行ファイルが出来上がるまでの流れは図のようになる。

プログラマはテキストエディタでソースファイルを書き，Cコンパイラを呼び出すだけだが，CC（GCC）ドライバは上に述べたようなたくさんのプログラムを次々と呼び出し，上に述べたようなファイルを続々と生成するのである。

図　ファイルの流れ

XCを支援するおいしいツール

ソースコードデバッガを使ってみよう

Izumi Daisuke 泉 大介

大きな魅力を持つC言語ですが，コンパイラであるがゆえ，インタプリンタのようにデバッグが手軽にはいきません。ところが，世の中には便利なツールがあるもの。XC ver.2.0の秘密兵器，それがこのソースコードデバッガです。

C言語でプログラムを作成するにはまずED.Xなどのエディタでプログラムを書き，それをファイルに保存します。そのファイルをXCでマシン語に変換させ，やっと実行できるようになるのです。

しかも，通常は一発でマシン語プログラムへの変換が終了するわけではありません。人間ですからミスをします。修正するには再びエディタを使ってコンパイラが間違っていると指摘する行を表示し，プログラムを直さなければならないのです[1]。

エラーはまとめて出力され，最初は山のようなエラーメッセージと格闘することになるでしょう。エラーのなかでも特にセミコロンの付け忘れは，信じられないようなエラーメッセージをもたらします。図1はリスト1の4行目のセミコロンを取ってコンパイルしたときに表示されるエラーメッセージです（Ver.2.0にてコンパイル)。このように多くの場合セミコロンのない行の次の行番号が表示されます。しかも，セミコロンがないとは指摘してくれません[2]。

止められない，止まらない

無事コンパイルが終了すれば，～.Xというマシン語ファイルが作成されます。が，ちゃんと動くという保証はどこにもありません。Cコンパイラは字面のチェックを行うだけで，実際に動かしたときにどうなるのかまでは面倒をみてくれないのです。

作成されたマシン語ファイルを動かして動作のチェックを行っていくのですが，ここで第2の関門が待っています。X-BASICならプログラムの実行中にBREAKし，変数の内容を表示してチェックできます。また，規定外の値を使用した場合には自動的にエラーで実行が中断されます。ところが生成されたマシン語ファイルの実行では途中で中断することはできませんし，規定外の値を使ったからといってプログラムが止まることもありません。

プログラムには，文字入力などBREAKキーが効く場所も存在します。しかしこの場合，BREAKした途端にプログラムの実行は終了してしまいますので，BREAKして変数の内容を確かめることはできません。変数の内容を調べたいと思ったら，怪しそうなところに変数を表示するプログラムを書き込み，再びコンパイルして実行する。これが従来行われてきたもっともオーソドックスなチェック方法です。さもなくば，デバッガ（DB.X）を使うしかありません。これは変換されたマシン語を対象としますので，マシン語の知識とCがどのようなマシン語を生成するかを知らなければにっちもさっちもいかないという代物です。

C言語で書いたプログラムを見ながら，自由に実行を中断したり変数を見ることができれば便利なのに……と，誰もが思うことでしょう。

ソースコードデバッガSCD.X

XCではVer.2.0からSCD.Xというソースコードデバッガが標準で付属しています。ソースコードデバッガを使うには，専用の情報をコンパイル後のマシン語ファイルが持っていなければなりません。Ver.2.0のXCでコンパイルするときに，

　　cc /Ns test.c

と「/Ns」スイッチを指定すれば，SCD.X用の情報をマシン語ファイルに持たせることができます。生成されたマシン語ファイルはそのままでも実行可能ですので，プログラム作成中はこのスイッチを指定しておくことにすればいいでしょう。ではリスト1でSCD.Xを試してみることにしましょう。

●プログラムのコンパイル

ED.Xなどを使ってlist.cを入力したら，

　　A>cc /Ns list1.c

としてコンパイルしてください。

　　X68k XC Compiler……

と表示されてコンパイルが始まります。間違いがなければそのまま何事もなく，

　　X68k Linker……

と表示され，最後に「A>」の状態に戻ればコンパイル終了です。

不幸にしてエラーが出たときには，

　　A>cc /Ns list1.c > err

として実行してみてください。今度は画面にエラーメッセージを表示する代わりにerrというファイルが作成され，この中にエラーメッセージが収められます。そこで，

　　A>ed err

として，このファイルをED.Xに読み込みます。最初のエラーメッセージでカーソルが点滅していますね。ここで「ESC」「V」の順にキーを押すと，あら，不思議。エラーが出たファイルを自動的に読み込み，該当行にカーソルが飛んでいきます。もう一度エラーを見たいときには「ESC」「A」と入力します。以後「ESC」「A」

1)　最近のCコンパイラには，エディタ，コンパイラ，そしてあとで説明するソースコードデバッガを統合したものが存在します。コンパイル時にエラーが発生すると該当行の該当箇所にカーソルが飛んでいき，すぐさま修正できるという環境が実現されているのです。手慣れたエディタではないという不満はあるものの，この便利さはうらやましいものです。

2)　最近になってようやく一部のパソコン用Cコンパイラでセミコロンがないというメッセージが表示されるようになってきました。

図1　エラーメッセージの例

```
list1.c   5 :Error  22:declaration error.
list1.c   5 :Error  56:compound statement error.
list1.c   6 :Error   9:external definition error.
list1.c  10 :Error  61:operand type mismatch.
```

リスト1　コマンドラインの引数を表示する

```
 1: #include <stdio.h>
 2:
 3: void main( argc, argv )
 4: int   argc
 5: char  *argv[];
 6: {
 7:   int   i;
 8:
 9:   for ( i=0; i<argc; i++ )
10:     printf( "%d:%s¥n", i, argv[ i ] );
11: }
```

と押すたびにエラーメッセージとlist1.cが切り替わって表示されます。

プログラムの修正がすんだら再びコンパイル。この作業をエラーが出なくなるまで繰り返します。

●さぁ，SCD.Xの出番だ！

コンパイルが終了したらSCD.Xを使ってみましょう。フロッピーディスクで使っている方は，ここでシステムを起動したディスクを再びAドライブにセットし直してください。SCD.Xは起動ディスクのほうに入っています。

準備が完了したら，

A>scd list1.x

でソースコードデバッガの起動です。写真1のような画面が表示されたことと思います。

一番上にはメニューが並び，2番目のウィンドウにはCのプログラムが，3番目にはCPUの変数（レジスタと呼ばれている）が表示されています。4番目がコマンド投入ウィンドウと呼ばれているもので，上級者になるとマウスでプルダウンメニューを開く代わりにここでコマンドを入力して作業します（彼らにとってはそのほうが速いらしい）。一番下のなにも表示されていないところはユーザースクリーンと呼ばれ，実行するプログラムが使用する画面です。マウスでクリックするとバーが現れます。いずれのウィンドウもバーを上下方向にドラッグすることによって大きさを変更できるようになっています。

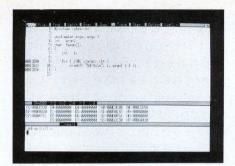
ではlist1.xを実行してみましょう。実行の前に画面下の黒い部分で左ボタンをクリックしてユーザースクリーンのタイトルを表示させ，1行分上にドラッグしておくといいでしょう。SCDはプログラム実行中にユーザースクリーンとSCDの画面を切り替えますので，起動時の状態では画面がチラチラして見づらくなるからです。ユーザースクリーンのタイトルを移動しておくとこの切り替えは行われません。

Execメニューから Runを選ぶとプログラムは通常のスピードで実行されます。これでは面白くありませんから，次のSlowを選んで実行開始です。プログラム中の1行が網掛け表示され，それが下へと動いていったのがわかりましたか？　SCDは網掛け表示で現在実行中の行を示すようになっているのです。実行が終了すると

program terminated normally

とコマンド投入ウィンドウに表示されます。プログラムは問題なく終了したという意味です。

写真1

写真2

ソースコードデバッグの実際

リスト1のプログラムはコマンドラインで起動時に与えた引数を表示するものです。たとえば，

A>list1 a b c

と入力すると，

0　A：¥list1.x
1　a
2　b
3　c

と画面に表示します。SCDでも引数を与えて実行させることが可能です。これはコマンド投入ウィンドウで行います。

－c a b c

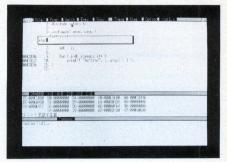
と入力してください。最初の「－」はSCDが表示したプロンプトですから入力の必要はありません。こうしてからプログラムを実行すると上のようにコマンドラインで引数を与えたのと同じことになります。

●変数を表示する

今度はただ実行を黙って見ているだけではなく，変数がどのように変わっていくかを追いかけてみることにしましょう。リスト1で使っている変数の中から，

i：ループカウンタ
argc：引数の個数
argv：引数配列

の3つを見てみます。変数を表示するにはWatchプルダウンメニューからSetを選択します（写真2）。以後この操作をWatch-Setのように略記します。変数名を入力するウィンドウが開きますから，まずiと入力してリターン。再びWatch-Setメニューでargcをセット。最後に「argv[i];s」とします。argvだけはi番目のものを文字列として表示することにしました（;sで文字列として表示することを示す）。

プログラムは2度続けて実行することはできません。Exec-Restartメニューで実行前の状態に戻し，再びExec-Slowで実行してみてください。表示されている変数がどんどん変わっていきますね。

●1命令ずつ追いかける

Slow実行でもまだ速くて変数の変化についていけないかもしれません。こんなときには1命令ずつ実行するTraceメニューです。Exec-Restartで実行前の状態に戻したら，今度はTraceをクリックしてみましょう。画面の1行が網掛けになり，変数が変わりました。以後Traceをクリックをするたびに1行ずつ実行されていきます。変数の変化を追うにはもってこいですね。

このプログラムを実行している間，ユーザースクリーンのほうはどう変化しているのでしょうか。画面を切り替えてみることにしましょう。まずメニューバーの右端にある□印のところをクリックし，SCDが全画面を占めるようにします。続いてShow-Screenメニューを選択するとユーザー画面が表示されます。ここでマウスのボタンかキーを押すと，元のSCDの画面に戻ります。先に説明したように，プログラム名とSCDのcコマンドで与えたパラメータが表示されていますね。ユーザースクリーンのタイトルバーを上にスクロールさせた状態ではSCDの画面は消えません。□印をクリックするのを忘れないでください。

SCDでCの勉強を

リスト2は九九の表を画面に表示するプログラムです。今度はこのプログラムを実行してみることにしましょう。

A>cc /Ns list2.c

でコンパイルしたら，リターンキーを押し続けてカーソルを画面の最下行までもって

リスト2　掛け算のプログラム

```
 1: #include <stdio.h>
 2:
 3: void main()
 4: {
 5:     int    i, j;
 6:
 7:     for ( i=1; i<10; i++ ) {
 8:         for ( j=1; j<10; j++ )
 9:             printf( "%4d", i * j );
10:         printf( "¥n" );
11:     }
12: }
```

写真3

いってください。そこで，

　　A>scd list2.x

としてSCDを起動します。今度は画面の下半分をユーザースクリーンにして実行してみましょう。画面下の黒いところでマウスの左ボタンを押すと表示されるタイトルバーを，そのままドラッグして画面中央まで引き上げればOKです。レジスタ表示はとりあえず必要ありませんので，Show-Registerメニューを選択して消去します。表示する変数はｉとｊです。ではTraceで実行してみましょう（写真3)。

　このプログラムではforの中でforが使ってある，いわゆる２重ループになっています。Traceするとまず７行目が網掛け表示されｉには１がセットされました。ｊには意味のない値が入っています。続いて８行目が網掛け表示され，ｊにも１がセットされます。そのあとは９行目から網掛け表示は動かず，変数ｊが１ずつ大きくなっていきます。ユーザースクリーンにはprintfを実行するたびに掛け算の答えが表示されていきますね。ｊが10になると内側のforの条件を満たさなくなり，網掛け表示は10行目に移ります。ここでユーザースクリーンは改行し，11行目が網掛け表示になったあと，再び８行目に戻ります。今度はｉがひとつ大きくなって繰り返しです。こうして１行ずつ実行していけばプログラムがどういう順序で動いていくのか一目瞭然です。途中までTraceしたら，あとはExec-Slowで一気に実行してかまいません。

　プログラムの動きを目で見ながら，画面に表示されていく文字を追いかける。こんな芸当はX-BASICにもできません。他人の書いたプログラムを読むのはなによりの勉強だといわれますが，雑誌に掲載されたリストを見る際にはどの変数にどんな値が入っているのかを覚えながら見ていかなければならず，面倒なことこの上ない。SCDならこの作業を全自動でこなしてくれるのです。見たい変数があればWatch-Setで加えるだけ。実に簡単です。

「初心者の方にはリストを読むことをお勧めします」これまではこういわれてきました。しかしSCDが登場した今となっては，「初心者の方はとにかく入力し，SCDで追いかけてみることをお勧めします」といったほうがいいでしょう。Ｃプログラミングも，SCDの登場によってプログラム実行時のチェックに関してはX-BASICを凌駕したといえます。

自作派に勧めるSCD

　SCDはプログラムを１命令ずつ実行するだけでなく，目的の場所まで一気に実行することが可能です。実行を停止する場所には若干の制約があり，Ｃのプログラムの左に16進数が表示されている行しか指定できません。これは，この行に対してマシン語が生成されているという印で，Exec-SlowもTraceもよく見れば左に16進数を従えた行だけを実行していくのがわかると思います。ここではちょっと高度な使い方になりますが，自作プログラムをデバッグするのに有利な機能を紹介しておきましょう。

●関数をスキップしてTrace

　Traceで１行ずつ実行すると，関数呼び出しがあった場合には呼び出された関数の中も１行実行の対象となります。これに対し，呼び出された関数を高速実行するのがStepです。網掛け行は次の行へ移動しますので，現在注目している関数だけを１行実行しているという感じになります。デバッグが終わった関数までいちいち実行されるのは……，という場合にお勧めです。

●指定場所でプログラムを停止

　任意の場所までプログラムを実行して停止させるのは簡単で，止めたい場所にマウスカーソルを移動し，右ボタンをクリックするだけです。簡単なプログラムを次第にバージョンアップしながらプログラミングを進めていく場合には，追加・変更した関数が呼び出されるまでは高速実行したいものです。このような場合に重宝する機能です。

　Ｃでプログラムを作っていると，勘違いから結構アドレスエラーやバスエラーを引き起こしてしまうものです。ちょっと力のある人ならすぐさまDB.Xを起動し，バスエラーを起こした場所を突き止め，さらにpsコマンドでシンボルを表示させて，それがどの関数内で起きたのかを調べることでしょう。そのあとはどうしますか？　再びエディタを起動し，プログラムとにらめっこというのが一般的なところでしょう。

　SCDなら問題の関数まで一気に実行し，そこからTraceすることで簡単に問題のある行を見つけ出すことができます。そもそもこれらのエラーが発生した場所をShow-Assemblyメニューで逆アセンブルリストを表示して確認することができるので，psコマンドがどうしたこうしたという面倒もありません。逆アセンブルリストを逆スクロールさせていけば，関数名のシンボルを見つけ出すのは簡単です。

●プログラム停止の条件を指定

　負の数を取らないはずの変数が－１になっているなど，変数が規定外の値を取っている。これもよくあるバグです。SCDはプログラムの実行を中止する条件を指定しておくことができます。Watch-WatchPtを選ぶと条件入力用のウィンドウが開き，停止条件をＣの文法で記述します。たとえば，

　　x <＝0

という具合です。条件が成立すればプログラムは停止します。

　また，特定の範囲のメモリを監視することもできます。こちらはWatch-TracePtです。Ｃの配列は宣言した大きさを超えて書き込んでもエラーにはなりません。ところが大きさを超えた部分はほかの変数が使用している可能性がありますので，結果として誤動作してしまうことがあるのです。先の，変数の値がいつの間にか変わってしまうという原因が，範囲を超えたアクセスだったというのもよくある話です。

　これら２つの機能はExec-Slow，Trace，Step実行中しか働きません。Exec-Runで高速に実行しながらチェックしてくれるとありがたいのですが。しかたがないので，変数が書き換えられてしまう関数呼び出しを見つけ出し，その関数の中をExec-Slowで実行してみるという使い方になるでしょう。

　ソースコードデバッガは使わない。そう決めている，という人がいます。動いているのはＣのプログラムではなくマシン語なのだから，マシン語用のデバッガのほうが便利だというのです。そんな方に最後にひと言。SCDは従来のDB.Xを機能強化してソースコードデバッガに仕立てたものです。もちろんメモリダンプからレジスタの設定，アセンブル機能まで，あなたがデバッガに必要だと思っている機能はちゃんと用意されています。Ｃ言語レベルでデバッグできる範囲は見やすいソースコードで，もっと細かく調べたいときはマシン語レベルで，これひとつでデバッグ可能になっています。ほら，これならどうですか？

使って便利な外部コマンド

Human68kのシステムディスクにはBINというディレクトリがあり，さまざまな外部コマンドが入っています。これらは，いずれも～.Xの形の実行ファイルで，あとから次々と付け加えて，Human68kの機能を拡張することができます。

便利な外部コマンドは，Human68kのver.2.0やXCでもいくつか追加されており，プログラム開発に積極的なユーザーのBINディレクトリはきっとかなり大きなものとなっていることと思います。ここでは，Human68kのver.2.0やXCで付いてくる外部コマンドのなかから代表的なものを紹介しておきましょう。

なお，Human68k ver.2.0で付いてくるものには（Hu），XC ver.2.0に付いてくるものには（XC）をつけました。また，特に表記のないかぎり，

　　コマンド名　スイッチ　ファイル名（ワイルドカード含む）

として使います。

●CASE　（XC）

指定したファイル（ただし読み出し専用およびシステム属性のある場合を除く）名に含まれる半角の英文字を小文字（/L），または大文字（/U）に変更します。また/Dでディレクトリ名を対象とします。ファイル名が省略されたときはカレントドライブ/ディレクトリのすべてのファイルが指定されます。

具体的にどのような場合に用いられるかというと，PC-9801へのファイル転送の際などです。MS-DOSではファイル名に半角英小文字を指定することができない（使うと大文字に変換してファイル名がつけられる）のでHuman68kで小文字でファイル名を付けられたものはそのままではファイル名を見ることはできても読み込むことができません。そこであらかじめHuman上で，

　　CASE A：

などとしてファイル名を半角大文字に変換するのです。

●DRIVE　（Hu）（XC）

ドライブ名を交換します。またドライブ名を指定しない，あるいは1つだけ指定するとそのドライブの状態を示します。/Vオプションでボリュームラベルも見ることができます。

マシン語プログラムの開発中などでプログラムのなかにファイル名が組み込まれていてドライブを変更したいのにどうしてもドライブ名が変更できないことがあります。そのようなときに効果を発揮するのがDRIVEコマンドです。ファイル名を書き換えるのではなく疑似的にドライブを入れ換えることによって目標を達成することができます。

　　DRIVE A：　B：

これでAドライブとBドライブを交換することができます。

●PROCESS　（Hu）（XC）

現在使われているメモリが何に使われているかを知ることができます。また自分のマシンがどこからどこまでにメモリが配置されているかを知っていればメモリの残り容量をだいたい知る（つまりMEMFREEを使わなくてもメモリの空き容量がわかる）こともできます。

また，Humanではバージョン2.0からバックグランドでタスクを実行するためのファンクションコールがサポートされた（今までのコマンドがバックグランドで動くわけではない）ので/bオプションでバックグランドプロセスの情報の表示，/aで従来のPROCESSの機能に加えバックグランドプロセスの情報を加えて表示を行います。

表示される内容は，

```
開始　 終了　 長さ　 モード　ファイル名
006800 077FFF 071800 SUPER Human.sys
                    ⋮
0F3480 0F367F 000200 MALLOC
0F3690 17FFFF 08C970 USER A:¥BIN¥PROCESS.x
```

で，

　　開始：メモリブロックの開始アドレス
　　終了：メモリブロックの終了アドレス
　　長さ：メモリブロックの大きさ

がそれぞれ16進数で表示されます。またモードのSUPER/USERはそれぞれプログラムがスーパーバイザー・ユーザーモードで実行されていることを示し，MALLOCはその表示のされている直前のプロセスによって確保されているメモリを示します。直前に表示されたファイル名＋MALLOCがそのプロセスが実際に使用しているメモリの大きさということになります。そしてファイル名ですがプロセスの起動されたときのパス，ファイル名です。

●WHERE　（Hu）（XC）

ファイルの検索を行うコマンドでデフォルトではカレントドライブから指定したファイルのパス名を表示しますがスイッチとして/A(すべてのドライブから探してくる）/F（パス名とファイル名を表示する）を指定することもできます。

Human68kやMS-DOSなどではファイル管理の方法として階層化ディレクトリを採用しており，ディレクトリを作りいくつかのファイルをそこに格納するのが普通です。この方法ではファイルの整理がしやすくなる反面，あまり複雑に階層をつけてしまうとファイルを格納してあまり長い時間がたつと忘れてしまうこともままあります。

たとえば，ゲームの原稿をどこにしまったのか忘れてしまった，たしかファイル名はGAMEという名前であったのだが……などという場合，

　　A>WHERE /a /f GAME.*

としてやれば，

　　a：¥artic¥game.10a
　　a：¥artic¥game.10b
　　c：は準備ができていません

（Cドライブにはフロッピーを入れていなかった）という具合に表示されます。

●TREE　（Hu）（XC）

ディレクトリ構造を表示します。

特定のファイルを探すのではなく，どこに何があるのかドライブ全体の情報を知りたいときにファイル名の一覧表を表示させることができます。WHEREコマンド同様/fスイッチがあり，すべてのファイル名を表示することができます（残念ながら/aスイッチはありません）。

また，TREEコマンドではすべてのファイルを表示するので特にハードディスクやMOなどを使用するとそのファイルの数は膨大なものになります。そこでファイル表示を見やすくするためにリダイレクションして，

　　A>TREE /f >ファイル名
　　A>ED　ファイル名

としてエディタを使うか，あるいは，

　　A>TREE /f | MORE

とパイプして見るのが普通です。

●TERM　（XC）

通信には通信ソフトを使うのが一般的ですが，RS-232C経由でコマンドのやり取りを行うだけであれば，実は通信用ソフトを使わなくてもとりあえず読み書きを行うことができます。XCに付属のTERMコマンドを使う方法がそれでマニュアルには，

　　TERM……ターミナルモードに入ります。

と実に簡潔に書かれています。

　　A>TERM ログファイル名

とすることでRS-232Cとの通信が始まります。これでヘイズモデムがつながっているのならば，ATコマンドを使ってそのまま通信ができるわけです。

　　[F7] キーで表示内容のプリントアウト
　　[F8] で表示内容の記録
　　[F9] でコマンドモードへ
　　[F10] で終了します。

●MAKE　（XC）

ファイル保守ユーティリティで，BINディレクトリに入っていますが，外部コマンドの規模を越えた多機能なツールです。これについては中森氏の記事をご覧ください。

＊

最後に外部コマンドを使うコツなど。再帰的なようですが外部コマンドを使いこなすためのコツはやはりいろいろな局面でコマンドを使ってみることではないかと思います。

というのもこういったものをうまく使いこなせるかどうかの決め手は，マニュアルにある機能をまる暗記することにあるのではなく，これはどんな局面でどう使えばよいのかということを身体で覚えることにあるからです。いくら本を読んで理屈がわかっていても実戦経験がなければ戦っても勝てないのと同じことです（もちろん両方とも同じくらい実戦経験があれば理屈を知っているほうが有利だろうが）。

皆さんもぜひひともいろいろコマンドを使ってトライ＆エラーを繰り返してコマンドを使うノウハウを手に入れてみてください。

（古村　聡）

貴方のプログラミングを支援する
縁の下のプリプロセッサ

Nakamori Akira 中森 章

プログラマが一定の書式にしたがって書いたものを解釈してコンパイルできるようにお膳立てしてくれるのがプリプロセッサです。うまく使えば自分のスタイルによって，プログラムを簡潔に，そして効果的に記述することができるでしょう。

PASCALやFORTRANをよく知っている人がC言語を始めたときに一瞬つまずくのがプリプロセッサの機能です。そう，ソースプログラムの中の#includeとか#defineといった#で始まるあのオマジナイです。

大学時代私は先輩からポータブルCコンパイラのソースプログラムを見せてもらった（なんでもそれがC言語の標準的なプログラムらしい）のですが，その中の#ifdefの山にすっかりまいってしまいました。全然意味がわからなかったのです。そしてプリプロセッサの大切さを実感したのです。現在でもプリプロセッサをC言語のおまけ的な機能としてしか紹介してない参考書も多いようですが，本当にC言語を使いこなせるようになるためにはプリプロセッサ機能をしっかりと理解しなければなりません。しかし，あのK&Rの教科書でさえプリプロセッサ機能は4.11節でほんの少しだけ説明されているだけですから，その重要度をつい見落としてしまいがちです。

その結果が何かわけがわからないままプログラムの先頭には必ず，

　　#include <stdio.h>

を書く習慣に落ち着くのです。実はこんなオマジナイはほとんどの場合不要なのです。私も会社に入って先輩からそのことを教えられたとき，頭をトンカチで殴られたような衝撃を受けました（今までは17文字も余分にプログラムを書いていたのか）。C言語を勉強しているみなさんも，いつの日かそのことを知って大人になっていくのです。C言語のプリプロセッサ機能を単なるオマジナイとして終わらせたのではオモシロクありません。そこで，今回はこのプリプロセッサ機能の基本的な部分を紹介することにしましょう。

プリプロセッサとは

Cコンパイラは C言語で書かれたソースプログラムを読み込み，コンパイルしてアセンブリ言語のプログラムを出力します。これがCコンパイラの役割です[1]。このとき，コンパイルの段階はさらに大きく2つの部分に分かれています。それが前処理(プリプロセス）とコンパイルです。つまり，C言語で書かれたソースプログラムは前処理によっていったん加工され，その加工されたあとのプログラムが実際にコンパイルされるのです。

C言語のプログラムの中によく見受けられる#includeとか#defineといったオマジナイはこの前処理段階で解釈されソースプログラムの加工が行われます。そしてこの前処理を行うためのプログラムがプリプロセッサなのです。

多くのCコンパイラの処理系ではプリプロセッサとコンパイラ（の本体）は別のプログラムで供給されています。たとえば，UNIXのCコンパイラではプリプロセッサはcpp，コンパイラはccomであり，XCのバージョン1ではプリプロセッサはCCP.X，コンパイラはCC0.XとCC1.Xになっています。このようにプリプロセッサとコンパイラが分かれている理由は，プリプロセッサの提供するマクロなどの機能が結構強力であるため，プリプロセッサを単独にファイル変換プログラムとして利用する目論みがあるのでしょう。ただし，RUN/CやTurboCなどコンパイル速度を「売り」にしているいくつかのコンパイラではプリプロセッサとコンパイラは一体になっているようです[2]。

さて，プリプロセッサの役割について説明しましょう。プリプロセッサの機能を大まかにいうと

- **単語の置き換え**
- **条件付きコンパイル**
- **別ファイルの取り込み**

の3つです。このほかにも細々とした機能はあるのですが，特殊な目的でしか使われないので省略します。

これらの機能の詳細と意義を説明する前に，ここではプリプロセッサがどんなものかを知ってもらうために単語の置き換えの例を紹介します。リスト1を見てください。これがC言語のプログラムであるといったら何人の人が信じてくれるでしょう。先に述べたように，#defineはプリプロセッサへの指示ですし，/*と*/の間はコメントですから，リスト1のプログラムの本体は，

　　I think that the X68000～

の部分になります。これは単なる英文であって，プログラムではありません。しかし，このプログラムはCコンパイラでコンパイルすることができます。嘘だと思うならコンパイルしてみましょう。X68000が2重定義だという警告は出ますが正常にコンパイルできるはずです。そして，コンパイル結果を実行すると画面に，

　　こんにちわ，世界

とプリントされます。いったい何が起きたのでしょうか。秘密は#defineにあります。

リスト1

```
 1: /*
 2:         プリプロセッサ機能（イントロダクション）
 3:
 4:         これはなんでしょう
 5: */
 6: #define I                main
 7: #define X68000           {
 8: #define best             "こんにちわ、世界\n"
 9: #define computer         printf
10: #define in               )
11: #define is               (
12: #define that             )
13: #define the
14: #define think            (
15: #define this             ;
16: #define world            }
17: /*
18:         これは単なる英文に見えますか‥‥
19: */
20: I think that the X68000 computer is the best in this world
```

これは単語の置き換えを指示するプリプロセッサ命令なのです。この命令によって先の英文の中の単語が別の単語や記号に置き換えられ，正しいＣ言語のプログラムに加工されたのです（えっ，わかってたの）。

試しにどのような加工が行われたのか見てみることにしましょう。XCならば/Pオプション，GCCならば－Eオプションを付けてリスト１のプログラムをコンパイルしてみてください。XCの場合は拡張子が.pのファイルに，GCCの場合は標準出力（大抵は画面上）に加工結果（プリプロセッサの出力）が出てきます。これを見ると，ほうら，先の英文が，

```
main( ) { printf ("……") ; }
```

と変換されているのがわかるでしょう。これは正真正銘のＣ言語のプログラムですね。まあ，実際のプログラムではここまで極端な（ふざけた）置き換えは行われませんが，プリプロセッサはソースプログラムがＣ言語で書かれたプログラムでなくても機械的にそれを加工していることに注目してください。プリプロセッサはＣ言語の文法とは無関係なのです[3]。ただし，プリプロセッサによって加工されたあとのプログラムはＣ言語の文法に合致していなければなりません。

結局，Ｃコンパイラの前処理として使用されるプリプロセッサの役割とは，なんらかの目的（読みやすくする，二者択一すべき表現を同時に記述する，など）でＣ言語の文法をはずれて記述されたプログラムをＣコンパイラが理解できるように加工することといえるのです。と，この程度のことを基礎知識として理解したうえで，いよいよプリプロセッサの個々の機能の説明をしましょう。ただし，今回はページ数の都合で＃define，＃ifdef，＃includeの３命令に焦点を絞って説明します。

---

1) XCのバージョン２では直接オブジェクトコードを出力するが，内部的にはいったんアセンブリ言語のプログラムを作り，それをアセンブルしているだけ。
2) XCのバージョン２もプリプロセッサとコンパイラの一体型である。ただし，コンパイル速度はXCのバージョン１より２倍以上遅い。GCCで最適化付きでコンパイルするよりも遅い。
3) だから汎用のファイル変換（あるいはマクロ処理）プログラムとして利用できる。

みんな使っているマクロ機能

プリプロセッサの機能でもっともよく使用されるのが文字列の置き換え機能です。これは指定された単語を別の単語や記号に置き換える機能[4]です。単語を置き換えるためのプリプロセッサへの指示としては，

```
#define
```

を使います。この機能は先のリスト１のプログラムで体験済みですね。ただしプリプロセッサでは，リスト１のような単純な単語の置き換えだけでなく，もう少し賢い置き換え（引数を持てる）もできます。全体として＃defineはマクロアセンブラのマクロと同等の機能を持っていますから，単語の置換機能はプリプロセッサでもマクロ機能と呼ばれています。

図１に＃defineを使用するための形式を示します。図を見てわかるように＃defineにはオブジェクト形式と関数形式の２つの形式が存在します。

オブジェクト形式のマクロは単なる単語の置き換えです。次の例を考えましょう。

```
#define   TEISUU    10000
```

は置換リストが10000であるTEISUUというマクロを定義することを意味します。このときプリプロセッサが，

```
x=TEISUU ;
```

という文を見つけると，

```
x=10000 ;
```

という置換が行われます。

このようにオブジェクト形式のマクロは定数の値を指定するために使われることが多いようです[5]。定数の値をマクロで定義しておけば，プログラム中で使用している定数に変更が生じた場合，そのマクロを定義している１行を変更して再コンパイルすればプログラム全体の変更を簡単に行うことができます。また，値が変更されることがないと保証されている定数に関しても意識的にマクロ化されていることがあります。たとえば，ほとんどの人がプログラムの最初で無意識に取り込むstdio.hというファイル（ファイルの取り込みに関してはあとで説明）の中ではEOF（End Of File：ファイルの終わり）とNULL（Null：何もない）という定数が，

```
#define EOF  (-1)
#define NULL 0
```

というように定義されています。このマクロを使用してＣ言語のプログラムを，

```
if((c=getchar()) ! =EOF){……
```

とか，

```
while( *ptr++ ! =NULL){……
```

といった具合に記述すれば，その意図するところをよりいっそう明確にすることができます。

次は関数形式のマクロです。これは引数を持った単語の置き換えです。引数はＣ言語での関数の定義と同様に（　）内にカンマで区切ることで指定します。たとえば，

```
#define MAX(X,Y)((X>Y)? X : Y)
```

というマクロ定義を考えましょう。これは２つの引数Ｘ，Ｙを持つＭＡＸというマクロを定義することを意味します。置換リストは，

```
((X>Y)? X : Y)
```

ですが，この中のＸとＹはそのままMAXの引数であるＸとＹに対応して置き換わります。つまり，引数の値に応じて単語（この場合は関数呼び出しのように見える）の置き換わり方が違ってきます。たとえば，

```
a=MAX(1,2);
a=MAX(a,b+1);
```

という文は，それぞれ，

```
a=((1>2)? 1 : 2);
b=((a>b+1)? a : b+1);
```

というように置き換わります。この関数型マクロは簡単な関数をプログラムの中で展開してしまうような場合に使用します。先のMAXというマクロは，

```
MAX(X,Y)
int X,Y;
{
    return((X>Y)?X:Y);
}
```

というように関数として定義してもプログラムの動きとしてはほぼ同じです。しかし，マクロのほうが実際に関数の呼び出し処理がない分だけ高速です。さらに，関数として定義してしまうと引数として渡せるデータの型がintなりdoubleなりに固定されてしまいます。マクロならば単純に置換を行うだけですから引数のデータ型がなんであっても同一のマクロで処理できます。このように関数型マクロは関数にはないうま味を持っているのです。

リスト２にマクロ機能の例を示しましょう。プログラムの動作はこのプログラムをコンパイルして実行すればわかると思いますので省略します。リスト２ではYESとNOをオブジェクト型マクロとして定義し，C_INPUTとSELECTを関数型マクロとして定義してありますね。それぞれがプログラムの中でどのように展開されるか（置

図１　#define（マクロ定義）の形式

```
1)  オブジェクト形式
      #define  識別子  [置換リスト]
2)  関数形式
      #define  識別子（[仮引数の並び]）
                      [置換リスト]

                               [　]内は省略可能
```

き換えられるか）考えてみてください。

ところで，#defineによるマクロ定義は1行に書くのが原則ですが，1行に書ききれない場合は¥を付けることで定義を次の行に継続することができます。リスト2のSELECTのマクロ定義がその例です。また，マクロの呼び出し（使用）は1行に書く必要はありません。もともとC言語には行の概念がありませんからそんな制限があったのでは不便でなりませんね。実際，リスト2のSELECTは呼び出し側が複数行にわたっています。これがどのように展開されているのかコンパイラの/Pオプション（GCCでは−Eオプション）で見てみるのも楽しいでしょう。

4）ANSI Cでは置き換える対象となるのはプログラム中のコメント，文字定数，文字列以外の部分に限られる。XCではプログラム全体。

5）このようなマクロ（値に名前を付ける）はmanifest（マニフェスト）定数と呼ばれる。知ったかぶりするのに使おう。

お得な条件付きコンパイル

次は条件付きコンパイルです。これはソースプログラムの一部分をいくつかの選択肢に分けておき，条件にしたがってどれかひとつを選択するというものです。そのためのプリプロセッサ命令が#ifdef，#else，#endifです。これはあるマクロ（#defineで定義するやつ）が定義されているかいないかによって実際にコンパイルすべき部分を選択します[6]。このとき，マクロが定義されている場合に選択するための命令が#ifdef，マクロが定義されてない場合に選択するための命令が#ifndefです。

#ifdef命令の基本形式を図2に示します。基本形式1では#ifdefで参照するマクロが定義されている場合，#ifdefと#endifまでの間をコンパイルの対象とします（ソースプログラムに取り込む）。もしマクロが定義されていなければその部分を無視します。基本形式2ではマクロが定義されている場合は#ifdefと#elseの間をコンパイルの対象とし，マクロが定義されていない場合は#elseと#endifの間をコンパイルの対象とします。たとえば，

```
#ifdef __GNUC__
    #define MAXSIZE 1000
#else
    #define MAXSIZE 500
#endif
```

という記述は，__GNUC__というマクロが定義されている場合は，

```
#define MAXSIZE 1000
```

というソースプログラム（の一部）になり，__GNUC__というマクロが定義されてない場合は，

```
#define MAXSIZE 500
```

となります。また，#ifdef命令は入れ子にして使うこともできます。

#ifdef命令はプログラムを実行するマシンやコンパイラの差異によって，C言語のソースプログラムを書き分ける必要がある場合によく使用されます。#ifdef命令が参照するマクロは先に説明した#define命令によって定義するのが普通です[7]。しかし，Cコンパイラにはあらかじめ定義済みのマクロというものがあって，それを利用することもあります。たとえばXCでは，

```
Human
X68000
```

などのマクロが，GCCでは，

```
__GNUC__
mc68000
x68k
X68000
__human68k__
```

などのマクロが定義済みです[8]。これらの違いを利用すればコンパイラの違いに依存した部分を1本のソースファイル内に共存させることができますね。このほかに，コンパイル時にマクロの定義をすることもできます。XCの/D，GCCの−Dがそのためのオプションで，

/DOSK　　（あるいは−DOSK）

というオプションを付けてコンパイルすれば，ソースファイルの先頭に，

```
#define OSK
```

があるのと同じ効果を持たせることができます。また，マクロ定義で置換リストを与えたいときは，

/DSIZE＝100

というように＝のあとで指定します。これはファイルの先頭に，

```
#define SIZE 100
```

があるのと同じです。ただこの方法で指定できるのは，いわゆるmanifest定数だけのようです。

それでは#ifdefの実例をリスト3に示しましょう。リスト3は指定したフィボナッチ数列の項を計算するプログラムです。ここではVALUEとSIMPLEという2つのマクロ名が参照されています。

VALUEは計算するフィボナッチ数列の項数を定義します。コンパイル時に/Dオプション（GCCでは−D）でVALUEに値（項数）が設定されないと，第24項を計算するようになっています。その指定をリスト3のどの部分で行っているのかはわかりますね。

SIMPLEはフィボナッチ数列の計算方法を指定します。コンパイル時にSIMPLEが/Dオプションで定義されていると，単純な再帰呼び出しでフィボナッチ数列を計算します。SIMPLEが定義されていないと，同じ再帰呼び出しですが，もう少し複雑な方法でフィボナッチ数列を計算します。

リスト2

```
 1: /*
 2:         プリプロセッサ機能（その1）
 3:
 4:         マクロ機能（文字列の置き換え）
 5: */
 6: #define YES     'y'
 7: #define NO      'n'
 8: #define C_INPUT(mess,var) printf("%s? ",mess),scanf("%c",&var)
 9: #define SELECT(var,case1,mess1,case2,mess2,mess3)        ¥
10:         if(var==case1) printf(mess1);                    ¥
11:         else if(var==case2) printf(mess2);               ¥
12:         else printf(mess3)
13:
14: main()
15: {
16:         char answer;
17:
18:         C_INPUT("中森　章は好きですか(y/n)",answer);
19:         SELECT(answer
20:                 ,YES,"いやぁ、照れ臭いなあ。¥n"
21:                 ,NO ,"ガーン、悲しい・・・・。¥n"
22:                     ,"好きか嫌いかはっきりしてよ！¥n");
23:
24: }
25:
```

図2 #ifdef/#ifndef（条件付きコンパイル）の形式

●基本形式1

```
#ifdef 識別子
   ⋮
#endif
```

●基本形式2

```
#ifdef 識別子
   ⋮
#else
   ⋮
#endif
```

#ifndefは#ifdefの逆条件

こちらはかなり高速です。みなさんもVALUEやSIMPLEを適当に定義してコンパイルしてみて遊んでみてくださいね。

---

6) 条件付きコンパイルには，このほかに，条件式が真か偽かによってコンパイルする部分を選択する #if，#elif，#else，#endifもあるが，あまり使われない。
7) 関数型のマクロが定義されているかどうかで選択する場合は，#ifdefでは引数の部分を抜かして参照する。
　例）#define INC1(X) (X+1)
　　　#ifdef INC1
　　　　⋮
　　　#endif
8) マクロは名前が定義されているだけで値（置換リスト）はない。#ifdefでのマクロ定義でも置換リストがないものがあるが，大抵が#ifdefでの参照用である。

別のファイルを取り込む

さて，今度はファイルを取り込む機能について説明しましょう。プリプロセッサはソースプログラムの指定した位置に別のファイルを読み込むことができます。そのための命令が#includeで，使用するときの形式は図3のようになっています[9]。#include命令は，取り込むファイルの内容を#includeがある位置にそのまま書き写したのと同じ効果を持ちます。このため，ソースプログラムの行数を短くできるという利点があります。プログラムで必ず書くような共通の宣言などは#includeで取り込むほうがいいでしょう。

#include命令の最も一般的な使用方法はヘッダファイルの取り込みです。ヘッダファイルとは，ライブラリで提供されている関数のプロトタイプ[10]および戻り値，マクロの定義をしてあるファイルです。システムであらかじめ提供されるヘッダファイルには.hという拡張子が付いていて，必要に応じて#include命令で取り込めるようになっています。たとえば，プログラムの先頭でオマジナイのように書くことになっている

　#include <stdio.h>

は，入出力のための関数のプロトタイプ，FILEというデータ型の定義，いくつかのマクロの定義を取り込むためのものです[11]。

図3 #includeの形式

●形式1
　#include 〈ファイル名〉
●形式2
　#include "ファイル名"

リスト3

```
 1: /*
 2:         プリプロセッサ機能（その２）
 3:
 4:         条件付きコンパイル（ifdef 系）
 5: */
 6: #ifndef VALUE
 7: #define VALUE   24
 8: #endif
 9:
10: fibon(n)
11: int n;
12: {
13: #ifdef SIMPLE
14:         if( n<3 ) return(1);
15:         else return( fibon(n-1)+fibon(n-2) );
16: }
17: #else
18:         return( fibl(1,n,0,1) );
19: }
20:
21: fibl(indx,max,val,va2)
22: int indx;
23: int max;
24: int val;
25: int va2;
26: {
27:         if(indx>=max) return( va2 );
28:         else return( fibl(indx+1,max,va2,val+va2) );
29: }
30: #endif
31:
32: main()
33: {
34: #ifdef SIMPLE
35:         printf("単純な再帰：フィボナッチ数列の第 %d 項は ", VALUE);
36: #else
37:         printf("複雑な再帰：フィボナッチ数列の第 %d 項は ", VALUE);
38: #endif
39:         printf("%d です。¥n", fibon(VALUE));
40: }
41:
```

リスト4

（a）

```
 1: /*
 2:         プリプロセッサ機能（その３）
 3:
 4:         ファイルの取り込み
 5: */
 6: #include "mydef.h"
 7: #include "myfunc.c"
 8: main()
 9: {
10:         double x,sq0,sq1,sq2;
11:
12:         D_INPUT("浮動小数点",x);
13:
14:         sq0=sqrt(x);    /* ライブラリより */
15:         sq1=fsqrt(x);   /* myfunc.c  より */
16:         sq2=fsqrt3(x);
17:
18:         printf("sqrt (%6.1f)= %f vs %f¥n",x, sq0, sq1);
19:         printf("sqrt3(%6.1f)= %f¥n",      x, sq2);
20: }
```

（b）mydef.hの内容

```
 1: #include <math.h>
 2: #define YES     1
 3: #define NO      0
 4: #define D_INPUT(mess,var) printf("%s? ",mess),scanf("%F",&var)
 5: #define F_INPUT(mess,var) printf("%s? ",mess),scanf("%f",&var)
```

（c）myfunc.cの内容

```
 1: /*
 2:         浮動小数点のライブラリを作ってみた。
 3:
 4:         注意：このプログラムを float 型のみで書くと
 5:               ＸＣではおかしくなるよん。
 6:               ＧＣＣではＯＫなんだな。こわい、こわい。
 7: */
 8: double fsqrt(x) /* ニュートン法による平方根 */
 9: double x;
10: {
11:         int   i;
12:         double y0,y1=0.5*(x+1.0);
13:         for(i=0;i<100;i++){
14:                 y0=y1; y1=0.5*(y0+x/y0);
15:                 if(y0==y1) break;
16:         }
17:         return(y0);
18: }
19:
20: double fsqrt3(x)        /* ニュートン法による立方根 */
21: double x;
22: {
23:         int   i;
24:         double y0,y1=(2.0*x+1.0)/3.0;
25:         for(i=0;i<100;i++){
26:                 y0=y1; y1=(2.0*y0+x/(y0*y0))/3.0;
27:                 if(y0==y1) break;
28:         }
29:         return(y0);
30: }
```

それでは＃includeを使用する例をリスト4(a)に示しましょう。＃include命令で取り込んでいるmydef.hとmyfunc.cというファイルの内容は，それぞれリスト4(b)，リスト4(c)に示してあります。これはニュートン法によって与えられた実数の平方根と立方根を求めるプログラムで，平方根についてはライブラリのsqrt関数と値を比較するようになっています。これ以上の説明は不要でしょう。ところで，リスト4(b)のmydef.hの中でmath.hを取り込んでいますが，このような多重の取り込みも可能です。ANSI規格では最低8レベルの取り込みが保証されています。

---

9) XCではファイル名を<>で囲むか" "で囲むかによって別の意味を持つ。つまり<>で囲んだ場合はまず標準的なディレクトリ（環境変数で決められている）でファイルを探してからカレントディレクトリを探す。一方，" "で囲んだ場合はカレントディレクトリから探す。ただし，ANSIでは，" "で囲もうが，<>で囲んだ場合と区別されていない。

10) 関数の引数（仮引数）の数とデータ型の宣言をプロトタイプ宣言という。広義には関数の戻り値もプロトタイプに含まれる。これによってCコンパイラはプログラム中で呼び出される関数の引数の数とデータ型，戻り値をチェックし，違反があると警告またはエラーを通知してくる。

11) したがって，入出力関数を使わない場合はstdio.hの取り込みは不要である。また，入出力関数を使っていても，関数のプロトタイプ宣言をしないつもりならFILE型を使用してない限りstdio.hの取り込みは必要ない。ただ，FILE型を使わないように見えるgetcharとputcharは，FILE型を使用するマクロとしてstdio.hの中で定義してあることもあり，その場合はstdio.hを取り込まないと都合が悪い。XCではgetcharやputcharはマクロではない（マクロにすることもできるが）ため，stdio.hの取り込みが必要な場合はあまりない。

ちょっとだけANSＩ機能

バージョン2が発売になって，XCもよりANSI準拠になったようですから，ここでANSI規格で定められているプリプロセッサの機能を少し体験しておきましょう。ANSI規格に準拠するCコンパイラではあらかじめ__STDC__というマクロが定義されています（XCのバージョン2でも定義されている！）。このマクロを＃ifdef命令で使用すればANSI規格に準拠した機能とそうでない機能の使い分けをすることができます。

ANSI規格でもっとも興味深いのは＃という演算子です。これは＃defineによる関数型マクロの置換リスト内で使用することができ，マクロの引数に＃を付けると引数が文字列として加工されます。たとえば，

```
#define MOJI(X) (#X,X)
```

という定義があるとき，

```
MOJI(A+B)
```

という表現は，

```
("A+B",A+B)
```

と加工されます。すなわち，＃を付けた部分は引数の値を" "で囲んだもので置き換えられます。

それでは，この機能がなぜ嬉しいのでしょう。これはANSI規格で新たに規定された文字列の連結機能と関係があります。文字列の連結機能（これはプリプロセッサの機能ではありませんが）とは2つ以上の文字列を並べて書いたとき，それが1つの文字列と認識される機能です。たとえば，

```
"ABCD" "EFG"
```

は，

```
"ABCDEFG"
```

と同じ意味を持ちます。

この機能をマクロで利用すれば文字列の一部をマクロへの引数の値にしたがって変更することができるようになります。たとえば，

```
#define SUKI(DARE). #DARE "が好き"
```

というマクロ定義があるとき，

```
SUKI(海)
```

という記述は，

```
"海" "が好き"
```

と置き換えられ，これは

```
"海が好き"
```

と同じことになります。このような文字列の加工はprintf関数の書式指定などで威力を発揮するでしょう。

リスト5に＃演算子を使用したプログラムの例を示します。これはXCのバージョン1では正常にコンパイルできないので注意してください。リスト5で注目してほしいのはDISPLAYというマクロでの＃の使われ方です。これは式の値をプリントするためのマクロですが，引数として与えた式からprintf関数に渡す書式と値をそのまま作り出しています。たとえば，

```
DISPLAY(a+b,%d);
```

は，

```
printf("a+b" "=" "%d" "n", a+b);
```

と置き換えられて，結局これは，

```
printf("a+b=%dn", a+b);
```

と同じになります。つまり，引数として与えた式にしたがって，どういう式の値かを表示するようにできるのです。ANSI規格以前ではこのような芸当は不可能でしょう。

＊

C言語のプリプロセッサ機能の主なものを概観してきました。C言語の学習の中でプリプロセッサの機能は忘れ去られる傾向にあります。自分でプログラムを書く場合にはプリプロセッサをほとんど知らなくてもなんとかなるでしょう。しかし，他人の書いたプログラムを読みこなすためにはプリプロセッサの知識は必須です。また，自分のプログラムを読みやすく，効率よく書くためにもプリプロセッサは大いに役立つことでしょう。これを機会に読者の皆さんも自分なりのプログラミングスタイルを考えてみるのもいいかもしれませんね。

《参考文献》

1) 林晴比古，Cプリプロセッサ・パワー，日本ソフトバンク，1988年．

2) 平林雅英，ANSI C言語事典，技術評論社，1989年．

3) マーク・ウイリアムズ社(編)，ANSI C言語大事典，パーソナルメディア，1990年．

リスト5

```
 1: /*
 2:         プリプロセッサ機能（その4）
 3:
 4:         ちょっとANSIな機能
 5:         XCのバージョン1ではコンパイルできないのねんねん。
 6: */
 7: #define G_INPUT(mess,type,var)  printf(#mess "? "),scanf(#type, &var)
 8: #define DISPLAY(expr,type)      printf(#expr "=" #type "¥n", expr)
 9:
10: #ifdef __STDC__
11: char date[]=__DATE__ "/" __TIME__; /* 文字列の連結が起きる */
12: char fname[]=__FILE__;
13: #else
14: char date[]="コンパイル日時は不明です。";
15: char fname[]="ファイル名は不明です。";
16: #endif
17:
18: main()
19: {
20:         int a,b;
21:         printf("ソースファイル:%s¥n", fname);
22:         printf("コンパイル日時:%s¥n", date);
23:
24:         G_INPUT(a=, %d,a);
25:         G_INPUT(b=, %d,b);
26:
27:         DISPLAY(a+b, %d);
28:         DISPLAY(a-b, %d);
29:         DISPLAY(a*b, %d);
30: }
```

基礎知識からプログラミングへ

Cライブラリ利用の手引き

Tan Akihiko 丹 明彦

C言語がその威力を発揮できるのは1にも2にもライブラリのお陰である。特にXCの魅力はその豊富なライブラリにあるといってもいいくらいだ。まずはCコンパイラとライブラリの熱い関係を理解して，プログラミングに挑戦してみよう。

ライブラリ。英語で書くとlibrary。図書館とか蔵書とかいう意味だ。Cプログラミングにおけるライブラリも，ほぼそういう意味だ。

XCのライブラリは豊富である。本当に豊富である。IOCSやDOSからBASICまで，X68000のあらゆるサービスがCから利用できる。ついでにいうと，このライブラリはGCC（GNU C）からも利用できる。ほんの数行プログラムを書いただけで，高度な処理が可能になる。

C言語はほかの高級言語に比べて開発用の言語としてはかなり優位な立場にある。その理由のひとつはライブラリが充実しているということにあると思う。

ライブラリはいろいろな意味で，通常の高級言語にある組み込み関数とは趣を異にしている。

ライブラリとはなんぞや

さて，Cを使った人がまず確実に引っ掛かると思われる（自分を基準にしてはいけないのだが）箇所について指摘しておきたい。使ううえでは支障にならないことだが，とても大切なことである。そしてこれが，通常の（手続き型）高級言語とCとの間にある差を象徴的に表していると僕は見る。ことはCの設計思想にまで及ぶ問題なのだ。いきなり質問から始めよう。

printf( )はCの命令である。○か×か。

答えは「×」である。これは冷静に考えてみると，とんでもないこととは思わないだろうか。画面に文字や数字をプリントするというプログラミング言語においてはとても基本的な機能すらも，Cの言語仕様には定義されていないのだ（例のK&Rでは，printf( )を標準的な関数として，その仕様を書いてあるが，それも文法の仕様とは別のものである）。このへんにCの本質のひとつをかいま見ることができる。

Cコンパイラそのものでできることは，実はアセンブラとたいして変わらない。アセンブラに毛の生えた程度の演算や，簡単な制御構造。これがC言語のほとんどすべてである。ただ，配列や構造体を使えるようにしていることで，複雑なデータ構造を処理できる。このおかげで，アセンブラよりはるかに変数などの管理が楽になっている。乱暴だが，Cは構造化されているアセンブラの代替品ともいえるのである。

ではprintf( )とはなんなのか。ここではサブルーチンのひとつという答えを与えておこう。文字のプリントなどというものを仕様の中にきっちりとうたっているアセンブラなどあるはずもないが，それでも文字のプリントは現実に可能な処理である。適当なパラメータで呼び出せば要求どおりの文字を出してくれる，そんなサブルーチンを用意しておいて呼び出せばいい。Cでは，これをサブルーチンと呼ばずに関数と呼んでいる。

そうした関数は，Cコンパイラを入手するともれなくついてくることになっている。それがライブラリである。ユーザーがプログラム中にprintf( )という関数を使用したら，コンパイラはライブラリの中からprintf( )関数を引っ張り出し，ユーザーの書いたプログラムとくっつける。そうすることで，printf( )があたかもCの命令のひとつであるかのように振る舞うことができるのである。

ところで，関数はユーザーが定義することもできる。main( )をはじめとして，Cのプログラムは関数の集まりである。そして，Cにおいては，ライブラリに入っている関数もユーザーが書いた関数もほぼ同じ扱いを受ける。自分が書いた関数をライブラリに追加することもできる。これはほかの高級言語風にいえば，命令をユーザーが増やせるというのと同じことなのである。

とはいえ，そんな基本的な関数からいちいち作っていくのは骨の折れる仕事である。そこにライブラリの存在意義がある。Cの処理系には，必ずライブラリが付属していて，printf( )などのようなごくごく基本的な関数は，自分で作る必要はない。最初にもいったとおり，XCの場合は，およそX68000を使う上で必要になりそうな処理はすべてライブラリで準備してある。ライブラリがないCなんて，単なるダルマさんなのである。

ライブラリとヘッダと関数とマクロ

ライブラリ理解への道に，またひとつハードルが見えてきた。それはライブラリとインクルードファイルとの関係。インクルードとは，他のファイルをソースリストに取り込むこと。このへんもアセンブラの影響を強く受けているようではある。インクルード命令を上手に使うと，以前からあるソフトウェア資産を手軽に生かすことができる。それではまたも質問形式でいってみよう。

関数printf( )を使うには，ヘッダファイル“stdio.h”をインクルードすればよい。これは正しいか。

答えは，「嘘ではないが正確でもない」だ（なんじゃそりゃ）。関数printf( )は，stdio.hをインクルードしなくても使えるし，またインクルードしただけで使えるとは限らない。printf( )を利用するためには，もっと別の方面から攻めなくてはならないのだ（というほど大袈裟でもない）。

種明かしをしよう。すでにガイドマップで解説されているように，ヘッダファイルでは，単に関数の外部参照宣言しかしていない。stdio.hにはプログラムらしいプログラムは書かれていない。ヘッダはあくまでもヘッダ，そこに関数の本体は書かないようになっている。関数の本体はライブラリに入っている。だから，

```
#include <stdio.h>
```

とソースリストに書くのは，この場合，

```
extern int printf( );
```

と書くのとほとんど変わらない。

stdio.hをインクルードした時点では，コンパイラは「あ，このプログラムはprintf

( )をどこかから呼び出してきて使うんだな」と認識するだけ。コンパイルが進行し，リンクフェイズに達して初めてprintf( )の本体を探しにかかるのである。このときリンカは，プログラムがかつてstdio.hをインクルードしていたなどということは知らない。わかるのは，受け取ったプログラムのなかでprintf( )が外部参照になっているということだけ。だから，ライブラリからprintf( )を探し出して，メインプログラムと結合しなくてはならない。このときもしもライブラリからprintf( )を見つけ出すことができなかったり，ライブラリそのものがなかったりしたら，当然エラーとなって，コンパイルは中断される。

このへんも，C言語を使っていてアセンブラの香りを強烈に感じる部分である。

なお，先ほど「関数printf( )はstdio.hをインクルードしなくても使える」といったが，これはprintf( )がごく素直なタイプの関数だからである。Cの関数宣言では，整数（int）型のものに限り省略してよいことになっている。しかし，無用のバグの発生を避けるため，また，他の人がこのプログラムを見たとき困らないように，ヘッダファイルをインクルードする命令は面倒でも入れておくようにしよう。

XCのための開発環境

Cプログラミングにおいて，これほど大切なライブラリであるが，いったいどこにあって，どうやって利用するのだろうか。といっても気にする必要なし。Cの処理系をインストールするときに環境設定してしまえば，あとは忘れてしまってかまわない。必要に応じて思い出すなり参考書を見るなりすればよい。

Cコンパイラを買えばシステムディスクがついてくる（当たり前）。そのディスクのディレクトリ構成を調べたりautoexec.batをよく読んだりすれば，設定のしかたはすぐにわかる。

システムディスクがAドライブだとしよう。ハードディスクでもフロッピーディスクでもかまわない。そこにライブラリやインクルードファイルを格納しておく専用のディレクトリを作る。その名もストレートにA：¥LIBおよびA：¥INCLUDE。

次に，Cコンパイラがライブラリやインクルードファイルを探しにいくときに**参照する環境変数**がある。その名もこれまたストレートでlibおよびinclude。これらには，ライブラリやインクルードファイルのあるディレクトリを指定する。値をセットするのにはautoexec.batを使うのが一般的である。

autoexec.batに，

set lib＝A：¥LIB
set include＝A：¥INCLUDE

の2行を加えておけば，以後コンパイラは自動的にライブラリやインクルードファイルのあるディレクトリを探しにいく。つまり，システムを再構築するまでは忘れていてもいっこうにかまわないというわけ。

ライブラリの選択

しつこいようだがXCのライブラリは豊富である。それだけにマニュアルも分厚い。コンパイラがバージョン2になって，ライブラリマニュアルが2冊に分かれた。それでも相当な厚さである。それくらいライブラリは豊富に取り揃えてある。だからといって怖がる必要はまったくない。まさか，これを全部暗記しないとCをマスターできないのでは？　と考える方はいないだろう。何度もいうように，ライブラリはCの文法そのものではない。あくまで，とても便利なサブルーチン群なのだから，必要なものだけつまみ食いしていくのが正しい。

多すぎて使いきれないライブラリ関数のなかでも，最もよく使われていると思われるものは，なんといってもstdio.h（で宣言される関数）であろう。スタンダードI/O，標準入出力の略だ。これとまったく関わりのないCのプログラムを探すほうが難しいくらいだ。とりわけprintf( )関数およびその類似関数と縁もゆかりもないプログラムは，めったにお目に掛かれるものではない。というわけで，stdio.h（で宣言される関数）の掌握は必須課題である。

あと，ちょっと細かい話になるし，たいして重要でもないが，コンパイルをつつがなく終わらせるために必要なことが少しある。

XCライブラリの関数には，レベルがつけられている。関数に格がついているのだ。レベルは0から3まである。原則的には，

レベル0：IOCS
レベル1：DOS
レベル2：C
レベル3：BASIC

となっている。レベルの数字は低水準の関数では低く，高水準になると上がるようである（ここでいう低水準とは必ずしも程度の低いことを意味しない。むしろ使う側に高度なものを要求する。一般に低水準関数のほうが使うのが難しく，それだけに高度な処理もこなせる)。これらのレベルの関数はそれぞれ，

IOCSLIB.A
DOSLIB.A
CLIB.A
BASLIB.A

というファイルに収められている。なお，XC ver.2では少々違っていて，ファイル名は～.Aではなく～.L，ファイル構造も異なるので互換性はない。

Cの標準関数はレベル2となっている。他の処理系への移植を考えるなら，レベル2の関数だけを利用すべきである。ハードに密着していたり，画面や音源まわりを直接ドライブするような危険なものでないならば，たいていの処理はレベル2で十分こなせる。例のstdio.hで宣言する関数群もこのレベル2の関数である。

XCでは，CLIB.Aに収められている標準関数以外の関数を使うときには，コンパイル時に/Wや/Yといったスイッチを指定する必要がある。これらのスイッチを使わないとコンパイルできないプログラムは，ほかの処理系へ移植するのが難しいと思って差し支えない。

関数利用の注意点

さて，ひとつの関数を利用するときに，調べておいたほうがいいことはいくつかある。

・その関数の機能が本当に自分の求めるものか

これは当たり前。関数の機能がよくわからないときは，とりあえずコンパイルしてみて，うまくいかなかったら別の関数にしてみる，という行き当たりばったり方式でもいい。ライブラリ関数のなかには，似たような機能を持ったものも多い。関数のレベルが違っていたりするので，安全そうなものを選ぼう。XCでコンパイルする場合，IOCS，DOS，それにBASIC関係のライブラリを利用するときにはコンパイル時にスイッチ/W(BASIC)，/Y(IOCS, DOS)が必要である。

・宣言はどのインクルードファイルで行われているか

＃include命令は書く必要のない場合も多いが，デバッグや移植のことを考えると，なるべく書くようにしたほうがいいのはいうまでもない。また，インクルードファイルでは，定数や構造体の宣言をしていることも多い。たとえばよく使うstdio.hにしても，ファイル構造体FILEやエンドオブ

ファイルを示す定数EOFを宣言しており，インクルードしないままコンパイルすると，「そんなもの宣言してないぞ」と怒られる。というわけで#include命令はまめに書くようにしよう。

・関数の戻り値はなにか，引数の型はなにか，数はいくつか

関数呼び出しでは，このへんからくるトラブルがとにかく多い。Cコンパイラは引数のチェックをほとんどしない。関数プロトタイプ宣言してあれば，多少はするとはいうものの，とても「暴走しない」というレベルまで面倒みてはくれない。バスエラーやアドレスエラーが出たときは，まずこのへんを疑うのがよいだろう。とにかく型チェックは基本である。

これだけのことをチェックしたら，プログラムリストに関数呼び出しの式を書くのだが，これがまた面倒なことに，ライブラリマニュアルに書いてある書式をそのまま書き移してもうまく動かない。ライブラリマニュアルのページ構成は図1のようになっているのだが，この読み方がちょっとわかりにくいかもしれない。例はprintf( )とput( )。XCバージョン2のライブラリマニュアルから引用した。

実は，マニュアルに書いてあるのは，関数本体，呼び出される側の書き方である。呼び出すほうでは違う書き方をする。つまりマニュアルの記述を多少読み換える必要があるのである。たとえば，

```
printf("ただいま処理中です¥n");
printf("(x,y)=(%d,%d)¥n", x, y);
put( x, y, x1, y1, pat, 16*16*2);
```

のように書く。これを図1と比べると，マニュアルの読み換え方が多少なりともおわかりいただけたと思う。一応まとめておくと，以下のようになる。

・関数の型であるintやvoidは書かない。

これが必要なのは関数本体（とインクルードファイルでの宣言部）だけ。

・変数名は，ユーザープログラムでの名前を使う。マニュアルに載っている名前を使う必要はない。

これは，ローカル変数のなんたるかをご存じの方には無意味なアドバイス。

・[ ]で括られた部分の引数は，いくつ書いてもいい。

もちろん省略も可能。

・型が"void *"となっている引数には，どんな型のポインタを使ってもいい。

たとえばput( )関数の場合は，bufがその変数へのポインタであれば何型でもかまわない。unsigned charやunsigned shortのことが多い。

ライブラリマニュアルはCをある程度使い慣れた人が読めるように作ってあるので，初めのうちは少々戸惑うこともある。

もっとも参考になるのは，実はページの下のサンプルプログラムである。結局，習うより慣れろというのが一番いいのかもしれない。ああ陳腐な結論。

楽して実を取れ

要領のいいプログラマはライブラリを上手に使う。一見して複雑そうな処理も驚くほど短い行数のプログラムで実現できる。優れたタッチタイピストであれば，一瞬のうちにソースファイルを書き上げることも可能である。

ライブラリの欠点は，まずなんといっても最高速でないこと。いくらCが危険性の高い暴走できる高級言語だといっても，ライブラリが平気でそんなことをやっていては話にならないわけで，ひととおりの安全設計とある程度の汎用性は必要なわけだ。そうなると，速度の落ちるのは必定。どうしてもスピードの欲しい場合には，専用のルーチンを自分で書くことになるのだが，それまではおとなしくライブラリを利用しておけばよい。ま，アフターバーナーならともかく，レイトレーシングで画面にドットを打つのにわざわざスーパーバイザモードに入って描画速度を稼ぐのも間抜けな話ではある。

図1　ライブラリマニュアルの例

```
printf          レベル2
[書式]  # include <stdio.h>

        int printf( format_string[, arg...] );
        char *format_string;

[機能]  format_stringで指定した……（以下省略）

put             レベル3
[書式]  # include <graph.h>
        void put( x1, y1, x2, y2, buf, size );
        int x1;
        int y1;
        int x2;
        int y2;
        void *buf;
        int size;

[機能]  bufが示す配列に格納された……（以下省略）
```

応用編

ライブラリを使って外部コマンドを作る

X68000ユーザーのほとんどがふだん使っている環境といえば，コマンドシェル(COMMAND.X)であろう。少なくともCプログラマはそのはずだ。SX-WINDOWの環境，特にCの開発環境が整うのはもうちょっと先の話。Cコンパイラも，コマンドシェルでの使用を前提に考えている。

で，コマンドシェルの環境を充実させようというわけだ。特に力を入れたいのは操作性の向上だな。そのためのアプローチはいくつかある。たとえば，速度の向上（IOCS.Xなどを使う），ヒストリドライバの活用，バッチファイルの作成などである。

そして新しいコマンドの作成というのもある。これが本題。ライブラリを使いまくれば，簡単に新しい処理を行うコマンドをいくらでも作ることができる。極端な場合，ひとつの関数につきひとつの外部コマンドを作ってしまう。パラメータはコマンドラインから取り込む。バッチファイルから次々と呼び出せば，インタプリタに近いことも可能である。

というわけで今月の色物。自作の外部コマンドとバッチファイルで，インタプリタもどきを作る。

プログラムには，有名な「エラトステネスのふるい」を選んだ。素数を求めるアルゴリズムの代表的なものである。今回作った外部コマンドは以下のとおり。それぞれリスト1～5となっている。

●numbers.x

指定された区間の自然数を数列にして標準出力に出力する。たとえば「numbers 15」とした場合，「12345」と画面に出力される。

●null.x

標準入力から入力した数列が空かどうか調べる。もしも空なら，環境変数errorlevelに1をセットする。

●head.x

標準入力から入力した数列の先頭の数を取って標準出力に出力する。

●tail.x

標準入力から入力した数列の先頭の数を除く数列を標準出力に出力する。

●div.x

標準入力から入力した数列の1番目の数が2番目の数で割り切れるかどうかチェックする。割り切れない場合は環境変数errorlevelに1をセットする。

リダイレクションの魔術師

さて，前記のコマンド解説にはやたらと「標準入力」「標準出力」という言葉が出てくる。これはもういかにもstdio.hの出番だな。解説しよう。

通常のファイルを扱うプログラムは，コマンドラインで指定したファイル名を受け取って，それに対応するファイルをfopen( )で開き，fscanf( )やfprintf( )でアクセスする。そうしてもよかったのだが，いかんせんプログラムが長くなる。それを嫌って標準入出力を使ってみた。標準入出力とは，通常はコンソールのこと。人間が計算機を使う場合の多くは，キーボードから手で入力し，出力が画面に出てくるのを目で見ることになる。それが標準ということの意味である。

標準入出力の場合，ファイルのオープン/クローズは不要。よってそれにまつわるエラー処理も不要になり，行数が短くできる。scanf( )やprintf( )などがあるので入出力したい場合にも困らない。また，標準入出力を使うようにしたおかげで，ファイルの受け渡しにリダイレクションが使えるようになった。リダイレクションとは，本来ならキーボードである標準入力の入力元をファイルに指定したり，本来なら画面である標準出力の出力先をファイルに指定したりすることである。パイプ機能と組み合わせてフィルタとして使えるようにもなる。

ただし，フィルタとして働くプログラムをリダイレクション入力（もしくはパイプ入力）指定なしで実行すると，カーソルが点滅したまま実行が止まってしまう（ように見える）。これはもちろん，標準入力，つまりキーボードからの入力を待っているからである。リダイレクションからの入力がない場合，標準入力はキーボードからとなっている。で，この状態を抜けてコマンドラインに戻りたい場合はCTRL＋Z（CTRLキーとZキーを同時に押す）がよかろう。これはエンドオブファイルを表すコントロールコードで，標準入力にこれが入るとフィルタのプログラムは終了する。

また，リダイレクション出力指定のない場合，標準出力は画面であるから，処理した結果は画面に出てくることになる。

コンパイルと実行

さて，今回のプログラムは，標準入出力（実際はリダイレクトするファイル）を線形リストとみなしたリスト処理プログラムになっている。さっそくプログラムを打ち込んでコンパイルしよう。これ以上短くできないというところまで短くしたので，打ち込みにたいして時間はかからないと思う。特殊な関数もいっさい使っていないので，単に

　CC　~.C

とするだけでいいはずだ。

では，デバッグも兼ねて，動作テストをしてみよう。図2のアンダーライン部をコマンドラインから入力して，出力が正しいことを確かめてもらいたい。出力のうち改行はこの図のとおりにはいかないが，そのへんはよきにはからっていただきたい。それでは解説の必要がありそうなところをちょこちょことつまんでみる。

第3行：見慣れない記号“｜”がある。Cでは，これは「または」だったが，コマンドシェルでは全然違っていて，いわゆるパイプである。左のコマンドの標準出力を横取りして，右のコマンドに渡す働きをする。この場合は，numbers.xの出力“3 4 5 6 7 8 9 10”をhead.xに渡している。head.xはその先頭の数字を取るので“3”と表示されるのである（第4行）。

第7行：パイプを2段につないでいる。numbers.xの出力“3 4 5 6 7 8 9 10”をtail.xに通すと“4 5 6 7 8 9 10”になる。これはもちろん空リストではないので，環境変数errorlevelは0のままである。なお，記号“||”は，BASICでいうマルチステートメントで，リダイレクションとは関係ない。

第8行：numbers.xに与える2つの引数が同じである。したがって出力は“3”だけ。それをtail.xに通すと，先頭以外の要素がないので，当然空リストが出力される。するとnull.xのチェックに引っ掛かってerrorlevelに1が立ち，第9行のメッセージとなる。

第10行：div.xの入力に“con”というファイルをリダイレクトしている。これはコ

リスト1　numbers.c

```
 1: /* numbers.c */
 2: #include    <stdio.h>
 3: int main( argc, argv )
 4: int      argc;
 5: char     *argv[];
 6: {
 7:      int a, b, i;
 8:      sscanf( argv[1], "%d", &a );
 9:      sscanf( argv[2], "%d", &b );
10:      for ( i=a; i<=b; i++ ) printf( "%d ", i );
11:      return ( 0 );
12: }
```

リスト2　null.c

```
1: /* null.c */
2: #include    <stdio.h>
3: int main()
4: {
5:      int i;
6:      return ( scanf( "%d", &i )==EOF );
7: }
```

リスト3　head.c

```
1: /* head.c */
2: #include    <stdio.h>
3: int main()
4: {
5:      int i;
6:      scanf( "%d", &i );
7:      printf( "%d ", i );
8:      return ( 0 );
9: }
```

リスト4　tail.c

```
 1: /* tail.c */
 2: #include    <stdio.h>
 3: int main()
 4: {
 5:      int i;
 6:      scanf( "%d", &i );
 7:      while ( scanf( "%d", &i )!=EOF ) {
 8:           printf( "%d ", i );
 9:      }
10:      return ( 0 );
11: }
```

リスト5　div.c

```
1: /* div.c */
2: #include    <stdio.h>
3: int main()
4: {
5:      int a, b;
6:      scanf( "%d %d", &a, &b );
7:      return ( (a%b)!=0 );
8: }
```

図2　動作テスト

```
 1: B>numbers 3 10
 2: 3 4 5 6 7 8 9 10
 3: B>numbers 3 10 | head
 4: 3
 5: B>numbers 3 10 | tail
 6: 4 5 6 7 8 9 10
 7: B>numbers 3 10 | tail | null | | if errorlevel | echo "nullです"
 8: B>numbers 3 3 | tail | null | | if errorlevel | echo "nullです"
 9: "nullです"
10: B>div < con | | if errorlevel |echo "割り切れません"
11: 4 2 ^Z
12: B>div < con | | if errorlevel |echo "割り切れません"
13: 5 3 ^Z
14: "割り切れません"
```

※ ^Zはコントロールのこと。CTRL＋Zキーを押して入力する。エンドオブファイルを表す。

ンソールの略で，キーボードから数字を入力させる。それが第11行の“4 2”である。4は2で割り切れるので，errorlevelは0。

第13行：今度は5が3で割り切れないのでerrorlevelが1になり，割り切れないというメッセージをもらう（第14行）。

＊

コーディング上の細かい話をひとつしておこう。null.cやtail.cで，エンドオブファイルをfeof( )で見ずにscanf( )の戻り値がEOF（実際の値は－1。stdio.hで定義してある定数である）かどうかで見ているが，これには理由がある。ファイルの終わりにスペースや改行などの空白文字が残っているときに，feof( )はエンドオブファイルとは見ないが，scanf( )は（もうファイルに数字が残っていないので）エンドオブファイルと見てくれる。数列なのだから，空白文字は無視してほしいというわけ。

＊

リダイレクションには，上で使ったもののほかにも追加リダイレクションというものがある。

type _a >> _b

とすれば，ファイル_bの後ろにファイル_aの内容がつながる。BASIC風にいえば，

_b＝_b＋_a

のようなものだ。

もうひとつ，これはリダイレクションではないが，copyコマンドにはファイルを結合する機能もある。上と同じことは，

copy _b＋_a _tmp||copy _tmp _b

とすればできる（同じといっても，画面に出てくるメッセージは少々違う）。この2つは覚えておくと便利かもしれない。

COMMAND.Xはプログラミング言語!?

さて，デバッグもすんだリスト処理コマンドを利用して，そろそろエラトステネスのふるいを作ってみよう。エラトステネスのふるいの考え方を図3に示す。

素数とは1とそれ自身以外に約数のない数のことである（1は素数ではない）。最初に2以降の数列を作っておく。数列の先頭の数は常に素数である。先頭の数を素数リストに追加する。数列の先頭を取った残りから，先頭の数の倍数を取り除く。すると，残った数列の先頭もまた素数になる。これを繰り返す。

というわけで，リスト6のprime.batを打ち込んで実行していただきたい。先ほどのリスト処理関数群と同じディレクトリに置いて，できればRAMディスクで動かしたほうがいい。気が遠くなるほど遅いし，ディスクアクセスも頻繁だからだ。実行ファイル名はいうまでもなくprime（primeとは素数のことである）。遅いので，最初は，

prime 10

のように小さい数から実行してみることを

望ましいコマンドの書き方

Cはコマンドラインから引数を受け取れるようになっている。これがもう便利なのである。さすがにUNIX由来だけのことはある。ほかのプログラミング言語でコマンドラインから引数をこれほど簡単にもらえるものを僕はあまり知らない（アセンブラは別だが）。たいていはいきなりINPUT文かなにかで，「ファイル名を入力してください」ということになっている。以下，よりよい操作環境のためのコマンドに身につけてほしい条件を，順不同に列挙しよう。

＊

対話形式プログラムの最大の欠点は，バッチファイルの中に書けないことであろう。対話形式プログラムが走り出すと，そこでバッチファイルの処理が中断し，ユーザーのキー入力をいつまでも待っているからである。先ほどのINPUT文はその典型。有名どころとしては，少し前までのMS-DOS。起動するたびに今日の日付と時刻を尋ねてきていた。要するにAUTOEXEC.BATにDATEやTIMEと書いてあっただけのことなのだが，いまどき内蔵時計のバッテリーバックアップは当たり前なのだから、これはなかなかに凶悪な仕打ちである。リターンキーでスキップすればいいということを知らなかったら，起動するたびに日付と時刻を調べて入力しなおすなどという間抜けなことにもなりかねない。対話形式にする必要が特にないプログラムは，対話形式にしないほうがずっと使える。

逆に，ハードディスクのフォーマットのように，滅多に使うものではなく，また自動実行されると困るようなものは，むしろ対話形式のものがよい。誤操作でデータがごっそり消えてしまうのを防ぐことができるからだ。また，それ単独で「シェル」と呼べるもの，ひとつの環境を作っているものは，対話形式でもいい。かなり不満ではあるが，MENU.Xも用途次第ではまあ許せる。対話形式を避けるべきなのは，あくまでコマンドシェルから呼び出すことを前提にしているツール類の話である。

＊

複数の引数を取りうるとき，その順序は任意にできるのが望ましい。特にスイッチの処理は，書く順序が限定されていると使いにくいことこの上ない。ひとつのスイッチに複数のオプションを記述する（たとえば「－xvf」のように）ことができるとさらによい。

＊

ファイル名の受け取り方にも柔軟性がほしい。処理するファイルの拡張子がわかっているような場合，拡張子を省略したり，つけたり，どちらもできるとうれしい。

＊

引数が必要なのにつけなかったとき，またはスイッチなどの使い方を間違えたときには，コマンドの使い方を簡単にでもいいからプリントアウトするのが親切。MS-DOSのコマンドは，「××はできません」「文法が違います」といったエラーメッセージが出てきておしまいということが多かった。それならどうすればいいのか。アドベンチャーゲームではないのだ。

これがUNIXだと少々違う。UNIXは確かに無愛想なOSだが，コマンドの実行に失敗したときは使い方を教えてくれる。詳しいオンラインマニュアルもあるが，使わなくてもなんとかなる。ついでにいっておくと，Human68kのコマンドは使い方を教えてくれるものが多い。PC-9801をX68000と同じ感覚で使っていていちいち引っ掛かるのはこのへんにも原因があるのかもしれない。

＊

エラートラップにも細心の注意を払いたい。暴走するのは最低だ。特に，引数の個数をしっかりチェックしていないコマンドは，使い方を間違えるとたちまちバスエラーやアドレスエラーの洪水だ。また，引数1にはこれを，引数2にはあれを，……というプログラマの決めた仕様をそのままユーザーに押し付けてもいいが，そのときはマニュアルを充実させること。それよりもプログラマの都合を押し付けない心掛けがむしろ大切。

＊

フィルタになりうるもの，特にフィルタとして使うとおいしいものは，標準入力と標準出力をうまく使ってフィルタにすること。フィルタとしてでなく使いたい場合も考えられるものは，「フィルタになるスイッチ」を設けてもいいし，引数の指定のしかたで動作を切り替えてもいい。このタイプのプログラムとしてはMORE.Xが代表的。

DIR | MORE

とすればフィルタとして，

MORE test.c

とすればページ単位のTYPEとして働く。

＊

グラフィックRAMを使うもの，マウスを使うもの，その他ハードウェアに密接するものは，へたをするとユーザーを混乱に陥れるので要注意。典型的な例を挙げる。

グラフィックRAMをRAMディスクとして使っている場合には，グラフィックを勝手に使ってRAMディスクを破壊しないようにする必要があるが，かといって，RAMディスクをテンポラリのドライブにしかしていないなら，使えないのも困りもの。とくに，グラフィックを使うとわかっているプログラム（画像ロードなど）なら，起動した時点でグラフィックを使うというユーザーの意志はあるわけで，それでも「グラフィックRAMは使えません」の一点張りではあまりにお粗末。メモリを拡張していたらどうということはないが，512Kバイトはとても魅力的な記憶容量，最大限に活用したいものだ。コンパイルのときはテンポラリファイルの置き場に，実行のときはグラフィックRAM本来の機能になっていてほしい。

だから，強制的にグラフィックRAMを使う（内容を破壊してもいい）スイッチ，またはグラフィックRAMの使用状態をチェックするスイッチを用意すると親切だな。といいつつ，今回のサンプルではそれをさぼっていたりする。

以上，自戒をこめて，外部コマンド作成の心得をおしまいにする。

勧める。ちなみに，100まで求めたら，RAMディスクで6分ほどかかった（！）。

リストの解読は，読者の皆さんにお任せしよう。Human68kのマニュアルを読めばすべて書いてあることだ。バッチファイルなのに生意気にもインデントなぞして，高級言語のふりをしている。

まあ，この程度のプログラムはすべてCで書いたほうがはるかに簡単だし，高速である。この「コマンドラインインタプリタ」は，実際呆れるほど遅い。もう少し速かったら，ドライストンベンチマークやら，果てはレイトレーシング（！）までするつもりだったが，どうも無理のようだ。

しかし，あえてCで書くのは最低限の基本コマンドだけに抑え，できる限りバッチファイル用の命令でプログラムを書いたのだ。試みとしてはけっこう面白い。まず，変数をファイルで持ったのがいい。今回はリストがその変数にあたる。いきなり可変長のデータ構造をサポートしているのである。それにハードディスクで実行すれば，停電しても変数の値が消えない（？）。

バッチファイルならばトレースも簡単。オールCで書いた場合，ソースコードデバッガなしでデバッグをしようと思ったら，ソースにprintf( )などを挿入してコンパイルしなおさなくてはならない。デバッグが終わったときは，さっき入れたprintf( )を削って再コンパイルだ。極めて面倒。しかしバッチファイルなら，typeなどをちょっと挟むだけですむ。プログラムによってはそれをdumpにしてもいいし，findだろうがdbだろうが思いのまま。要するにアルゴリズムにあわせて好きなデバッグ用ツールを使い放題にできるのだ。デバッグが終わっても，その行を削除するだけ。要するに，基本コマンドさえ充実しておけば（重要），あとはインタプリタとほとんど同じ感覚で使えるのである。

BASICライブラリもお得

最後におまけとして，BASICライブラリを用いたコマンドサンプルをつけておく。ここで述べてきたことも多少は生かしているので，参考にしていただきたい。

簡単な画像処理コマンドである。名前はACCENT.X。65536色モードで画像をロードしておき，ACCENT.Xを実行すると，色が強調される。何度もエフェクトをかけると，しまいには原色（8色！）になって楽しい。約2年前（1988年9月号）の画像処理プログラムのうち，色強調処理を取り出して高速化した（といっても，XCでコンパイルした場合とGCCでコンパイルした場合とでは，速度に雲泥の差がある）。

コンパイルは，ACCENT.CがBASICライブラリを使っているので，次のように/Wスイッチをつける。

CC /W ACCENT.C

あとはPICでいろいろな画像をロードしては色鮮やかに変換してお楽しみいただきたい。短いプログラムなので手軽に作れてすぐ使える。ほんのちょっと改造すれば，白黒変換コマンドにもなる。

図3　エラトステネスのふるい

1:　　　2 3 4 5 6 7 8 9 10 11　始めの状態。
2:　2　←　3 ~~4~~ 5 ~~6~~ 7 ~~8~~ 9 ~~10~~ 11　先頭の2を素数リストに追加し、残りの数列から2の倍数を取り除く。
3:　2 3　←　~~4~~ 5 ~~6~~ 7 ~~8~~ ~~9~~ ~~10~~ 11　以下繰り返す。
4:　2 3 5　←　~~6~~ 7 ~~8~~ ~~9~~ ~~10~~ 11
5:　2 3 5 7　←　~~8~~ ~~9~~ ~~10~~ 11
6:　2 3 5 7 11　←　　結果

リスト6　prime.bat

```
 1: ECHO OFF
 2: REM エラトステネスのふるい
 3: IF NOT "%1" == "" GOTO START
 4: ECHO 使い方:PRIME [n]
 5: ECHO      n までの素数を求めます。
 6: GOTO END
 7: :START
 8: numbers 2 %1 > _n
 9: ECHO 素数: > prime.lst
10:   :LOOP1
11:   null < _n || IF ERRORLEVEL 1 GOTO RESULT
12:   head < _n > _h
13:   tail < _n > _t
14:   rem TYPE _h
15:   COPY prime.lst+_h _tmp  > NUL || COPY _tmp prime.lst > NUL
16:   COPY NUL _n > NUL
17:     :LOOP2
18:     null < _t || IF ERRORLEVEL 1 GOTO LOOP1
19:     head < _t > _a
20:     tail < _t > _t
21:     COPY _a+_h _b > NUL
22:     div < _b || IF ERRORLEVEL 1 COPY _n+_a _tmp > NUL || COPY _tmp _n > NUL
23:     GOTO LOOP2
24: :RESULT
25: DEL /Y _* > NUL
26: TYPE prime.lst
27: :END
```

リスト7　あると便利な色処理マクロcolor.h

```
1: #define     RED(C)       ( ((C)>>6 ) & 31 )
2: #define     GREEN(C)     ( ((C)>>11) & 31 )
3: #define     BLUE(C)      ( ((C)>>1 ) & 31 )
4: #define     RGB(R,G,B)   ( ((R)<<6) | ((G)<<11) | ((B)<<1) )
```

リスト8　画像の色を強調するaccent.c

```
 1: #include    <iocslib.h>    /* 関数 CRTMOD() のためのインクルードファイル */
 2: #include    <stdio.h>      /* 関数 fprintf() */
 3: #include    <graph.h>      /* 関数 get(), put() */
 4: #include    "color.h"      /* マクロ RED(), GREEN(), BLUE(), RGB() */
 5:
 6: unsigned short  pixel[ 512 ];
 7:
 8: void    main()
 9: {
10:     unsigned int    x, y;
11:     unsigned short  c;
12:     unsigned int    r0, g0, b0;
13:     int             r1, g1, b1;
14:
15:     if ( CRTMOD(-1)!=12 ) {
16:         fprintf( stderr, "画面が 65536 色モードになっていません。¥n" );
17:         fprintf( stderr, "65536 色モードにするには、SCREEN 1,3,1 とします。¥n" );
18:         fprintf( stderr, "そのうえで画像をロードしておいて実行してください。¥n" );
19:         fprintf( stderr, "画像の色を強調します。¥n" );
20:         return;
21:     }
22:     for ( y=0; y<512; y++ ) {
23:         get( 0, y, 511, y, pixel, 512*sizeof(short) );
24:         for ( x=0; x<512; x++ ) {
25:             c=pixel[x];
26 :            r0=RED(c);
27:             g0=GREEN(c);
28:             b0=BLUE(c);
29:             if ((r1= (r0*2)-(g0/2)-(b0/2))>31) r1=31; else if (r1<0) r1=0;
30:             if ((g1=-(r0/2)+(g0*2)-(b0/2))>31) g1=31; else if (g1<0) g1=0;
31:             if ((b1=-(r0/2)-(g0/2)+(b0*2))>31) b1=31; else if (b1<0) b1=0;
32:             pixel[x]=RGB(r1, g1, b1);
33:         }
34:         put( 0, y, 511, y, pixel, 512*sizeof(short) );
35:     }
36:     return;
37: }
```

多数のソースファイルを管理する

XCにMAKEが付いてきた

Nakamori Akira 中森 章

C言語では，プログラムのモジュール化によって大きなプログラムも効率的に作成されます。が，そのためには数々のファイルをきちっと保守管理しなくてはなりません。そこで登場のMAKE。XCの環境がまた一歩進みました。

C言語で巨大なプログラムを作成するときに必要になるもの。それは努力と忍耐，そしてMAKEです。

プログラム開発の基本は，プログラムをモジュール化して別々のファイルで作成し，最後に結合するのが一般的です。これがいわゆる分割コンパイルというやつですが，これはファイルの数が多いと結構煩わしい作業になります。たとえば，あるファイルを修正したり変更したりしたとき，その修正や変更がほかのファイルのプログラムにも影響を及ぼすものであるなら，影響されるファイルをすべて再コンパイルしなければなりません。多数のファイル間の依存関係なんて開発期間がちょっと長くなると忘れてしまいますからね。

そこでMAKEが必要になります。ファイル間の依存関係を気にせず自由にファイルを修正/変更する勇気を与えてくれるツールがMAKEなのです。XCのver.1ではコンパイルオプション（/M）でMAKEもどきなコンパイルを行うこともできましたが貧弱な機能しか持っていませんでした。ところが，XCのver.2.0ではとうとうMAKEが標準で付属するようになったのです。たかがコンパイラのおまけと侮ってはいけません。プログラマーズマニュアルのMAKEの解説を読めばそれがかなり本格的なものであるとわかります（馴染みのない人には理解できないかも）。

今回はMAKEの解説ということなのですが，豊富過ぎる機能をすべて紹介するのは不可能ですから，MAKEの初歩の初歩を説明してその有用性について知ってもらう程度にとどめましょう。

MAKEが必要なそのわけは

MAKEの機能をひと言でいえば，指定した手順に従って複数のソースファイルをコンパイルしたりリンクすることです。このときMAKEはあらかじめ与えられているソースプログラム間の依存関係を調べ，変更したソースファイルが影響を与える最小限のファイルだけをコンパイルしたりリンクしたりしてくれるのです。

たとえば，次のような例を考えてみましょう。プログラムprog.xは，Cのソースファイルmain.c, subr1.c, subr2.cをコンパイルして作られていると仮定します。さらにsubr1.cとsubr2.cは共通のヘッダファイルcommon.hをインクルードしているものとします。このときprog.xはXCでは，

```
cc /Fxprog main.c subr1.c subr2.c
```

を実行することによって作られますが，このやり方は，まだソースファイルのバグが取り切れてない状態では，どれかのソースファイルを変更するたびに3つのソースファイルをすべてコンパイルすることになって効率がよくありません。

コンパイル時間を節約するためには，

```
cc /Fc main.c
cc /Fc subr1.c
cc /Fc subr2.c
```

というように/Fcオプションによって，すべてのソースファイルをリンクする直前のオブジェクトファイルの形式（拡張子が.oであるファイル）で保存しておき，その後，

```
cc /Fxprog main.o subr1.o subr2.o
```

によってオブジェクトファイルをリンクしprog.xを作成します。

もし，subr1.cに変更があった場合は，

```
cc /Fc subr1.c
```

というように，変更のあったソースファイルのみをコンパイルしてオブジェクトファイル（subr1.o）を作り，再び全体をリンクすれば，新しいprog.xを作ることができます。この場合，変更のなかった2つのソースファイルをコンパイルする必要はありません。

次はcommon.hが変更された場合を考えましょう。これは，そのファイルをインクルードしているsubr1.cとsubr2.cの2つが変更されたのと同じことになります。したがって新しいprog.xを作るためには，

```
cc /Fc subr1.c
cc /Fc subr2.c
```

によってsubr1.cとsubr2.cをコンパイルして新しいsubr1.oとsubr2.oを作り，

```
cc /Fxprog main.o subr1.o subr2.o
```

によってリンクしなければなりません。これはひとつのファイルの変更が2つ以上のソースファイルに影響を与える例です。もし，subr2.cがcommon.hをインクルードしているのを忘れて，subr1.cのみしかコンパイルしなかったらリンク後のprog.xがまともに動くわけはありませんね。分割コンパイルを行うとこのようにソースファイル間の依存関係にいつも注意してなければならないのです。

上の例ではファイルの数が少ないので，頭の中で少し考えればある変更に対してどのソースファイルをコンパイルすべきかということはわかります。しかし，ファイルの数が多くなるとこんなにすんなりとはいきません。なによりも，変更された多くのソースファイルをコンパイルするためのキーボードからのコマンド入力は面倒です。

MAKEはこのような煩わしい手順から解放してくれるツールなのです。

ところで，定型的なコンパイル，リンクという作業を一括して行うことはHuman68kのバッチファイルを使っても可能です。しかし，バッチファイルではソースファイル間の依存関連を調べて必要なものだけをコンパイルするなどという芸当は不可能に近く，仮にユーティリティプログラムなどを駆使して実現できたとしても非常に複雑なものになってしまうでしょう。

簡単な手順でコンパイルやリンクを自動化することがMAKEの意義なのです。

注意）ここで説明したオプションはXC ver.2.0のもの。/FcはXCのver.1では/L，GCCでは－cである。また，/FxはXCのver.1では/Z，GCCでは－oである。

便利さの秘密は

MAKEは変更されたファイルが影響を与えるソースファイルを自動的に探し出し

ます。といっても，MAKEは単なるユーティリティプログラムにすぎませんからそこにはトリックがあります。それは，それぞれのファイル間の関係やファイルが変更されたときの動作を記述するMakefileというファイルです（もともとファイル名の大文字，小文字の区別はないのでmakefileでもMakeFileでもなんでもよい）。

MAKEはこのMakefileの内容を頼りにMAKE自身が行うべき動作を決定します。また，ファイルが変更されたかどうかの判断はコンパイルなどによって作られるファイルの作られた日時がそれの元になるソースファイルの作られた日時より前かどうかによって行います。したがってMAKEを使用する場合はあらかじめMakefileを書いておく必要があります。

それではMakefileの書き方について説明しましょう。Makefileでは基本的には，ソースファイルとそれをコンパイルやリンクによって作られるファイル（ターゲットファイルという），およびソースファイルからターゲットファイルを作るためのコマンド行を記述します。具体的には，

ターゲットファイル：ソースファイル
　　コマンド行

という記述です。ひとつのMakefileの中にはこの組み合わせが必要な数だけ記述されています。ここで，：(コロン)がファイルの依存関係を示しています。：の右側にあるソースファイルはいくつあってもかまいません。ここに必要なソースファイルを書き忘れると当然MAKEは正しく動きません。コマンド行には通常ソースファイルからターゲットファイルを作るためのコマンドを書きます（コマンド行はなぜかタブで始まらなければならない：重要)。コマンド行は続けて何行書いてもかまいません。このとき各コマンドが連続して実行されます。

これだけの知識でMakefileを書くことができます。たとえば，先に述べたmain.c, subr1.c, subr2.cからprog.xを作る場合にはリスト1に示すようなMakefileを書けばよいでしょう。最終的に作られるファイルの関係をいちばん最初に書くことを除けば，あとは適当な順序で記述してもなんとか動くでしょう。結構簡単だと思いませんか？ Makefileを書いたあとは，キーボードから，

MAKE

と打ち込むだけで，すべてのことをMAKEがやってくれるでしょう。

ところで，リスト1のMakefileは単純明快ですが，みなさんが目にするMakefile，たとえばPDSなどをコンパイルするときに使用するMakefileはもっと複雑でわけのわからないものが多いと思います。これはMAKEの提供する省略機能やマクロを駆使してMakefileが効率よく（読みにくいけどかっこいい）書かれているためです。どのように記述しようと効果は同じですが，Makefileをかっこよく書きたい人はマニュアルで勉強してくださいね。

注意）コンパイルオプションの差異から，リスト1のMakefileはXC ver.2.0でしか正しく動かない。XCのver.1やGCCで使用する場合はコンパイルオプションを変更する必要がある。また，この理由からUNIXで使用されているMakefileを持ってきても，そのままではまず正しく動かない。

リスト1

```
1: prog.x : main.o subr1.o subr2.o
2:         cc /Fxprog main.o subr1.o subr2.o
3: main.o : main.c
4:         cc /Fc main.c
5: subr1.o : subr1.c common.h
6:         cc /Fc subr1.c
7: subr2.o : subr2.c common.h
8:         cc /Fc subr2.c
```

リスト2

```
 1: /*
 2:         main.c   いわゆるメインプログラム
 3: */
 4: main()
 5: {
 6:         int x;
 7:         scanf("%d", &x);
 8:         printf("func1=%d¥n", func1(x) );
 9:         printf("func2=%d¥n", func2(x) );
10: }
```

リスト3

```
 1: /*
 2:      subr1.c
 3: */
 4: #include "common.h"
 5:
 6: func1(x)
 7: int x;
 8: {
 9:         return( CONST+x );
10: }
```

リスト4

```
 1: /*
 2:         subr2.c
 3: */
 4: #include "common.h"
 5:
 6: func2(x)
 7: int x;
 8: {
 9:         return( CONST*x );
10: }
```

リスト5

```
1: /*
2:      common.h  たったこれだけですが‥‥
3: */
4: #define CONST 10
```

MAKEを体験しよう

説明だけではおもしろくないので実際にMAKEを動かしてみましょう。Human68kのMAKEはMAKE.Xというファイルです。これをPATHの通じているどこかのディレクトリに置いておいてください。これが最小限の準備です。次にMakefileが必要ですが，これはリスト1のものを使いましょう。これに従って3つのソースファイルとひとつのヘッダファイルを用意します。これは何でもいいのですが，とりあえずリスト2〜5を使うことにしましょう。リスト2からリスト5のファイルとリスト1のMakefileをひとつのディレクトリに集めたら，

MAKE

と打ち込んでください。main.c，subr1.c，subr2.cが次々とコンパイルされリンクされてprog.xが作成される様子を見ることができます。これを確認したら，

1)　subr1.cだけを書き換えたあとにMAKEを実行する（MAKEと打ち込む)。
2)　common.hを書き換えたあとにMAKEを実行する。

と操作をしてみてください。1)の例ではsubr1.cだけがコンパイルされなおすこと，2)の例ではsubr1.cとsubr2.cのみがコンパイルされなおすことが確認できます。

注意）XCのver.1とver.2.0では標準的なヘッダファイルであるFEFUNC.Hに互換性がないので注意。FEFUNC.Hとコンパイラのバージョンが一致してないと正常にコンパイルできない。

＊

現在，UNIX上でのプログラム開発においてMAKEを使用することは半ば常識のようになっています。今回の解説でMAKEに興味を持った人は，是非とも実際にMAKEを使用してみることをお勧めします。特にソースプログラムが2つ以上になる場合は，今回紹介したごく初歩的な機能だけでも結構重宝するはずです。

それにしても，ソースコードデバッガも付属したことだし，XCもいよいよまともにプログラミングできる環境を目指してきたんだなというのが最近の実感です（バグもまだあるようだけど)。

MIDI制御が加わった
新しい音楽ドライバOPMDRV2.X

Nishikawa Zenji 西川 善司

新しいCコンパイラとともに突然現れたOPMDRV2.X。初めてのシャープ提供によるMIDIドライバの概要と使い方を見ていきましょう。MT-32を使用したサンプルプログラムも掲載します。参考にしてください。

私が駅前の新興宗教の勧誘によく捕まるモゲランチョ西川善司です。突然ですが，「C compiler PRO-68K ver. 2.0」を買うとバージョンアップされた音源ドライバ「OPMDRV2.X」がついてきます。ここでは，そのドライバが前の「OPMDRV.X」とどう違うのか，またマニュアルに載っているサンプルプログラムでは少しわかりにくいと思われるチャンネルアサイン関係について説明します。

拡張された命令たち

「OPMDRV2.X」になって拡張されたのはMIDI出力関係です。はっきりいってしまえば，FM音源関係についてはなにひとつ変わっていません。まぁ，以前のミュージックデータとの互換性のからみのため，変えようがなかったのでしょう（期待した人残念でした）。

その代わりといってはなんですが，MIDI制御用の新しいコマンドが追加されているのでそれをリストアップしてみます。

md_play
md_stop
md_cont
md_init
md_stat
md_on
md_off
md_regr
md_rdgw
md_wrt

表1　MIDI拡張MML一覧

MMLデータ	意　　味	パラメータ範囲
'(アポストロフィ)	MIDI拡張MMLの使用を開始/終了する	
Tn	MIDI送信チャンネルセット	1～16
Pn	MIDIプログラムチェンジ	1～128
On, m	ノートオン	0～127
Fn	ノートオフ	0～127
$n	ダイレクト送信データ	0～255
n	ダイレクト送信データ	0～255

（前記以外に既存命令と同名でMIDIに対応している，というのもあります）。

まず，上から5つは説明するまでもなく既存の命令「m_play」,「m_init」など……のMIDI版といった感じです。その下の「md_on」「md_off」は引数で与えたMIDIチャンネルへの出力をスイッチする命令です。そのまた下の2つ「md_regr」「md_rdgw」はX68000のMIDIコントローラ，YM3802からのレジスタ読み込み，またはレジスタへの書き込み命令です（ん～，マニアック！）。最後の「md_wrt」はMIDIの生データの出力命令で，まあ，これさえあれば理論上はなんでもできるというやつですな。

拡張されたMML

MIDI楽器をMMLでコントロールできるようにと「拡張MML」なるものが使用可能になりました。「拡張MML」は，従来の「OPMDRV.X」のミュージックデータと区別しやすくするためか「'」（アポストロフィ：[SHIFT]＋7）でくくられた中に記述します。既存の「FM音源専用のミュージックプログラム」をMIDI対応に改造するには任意の位置にこのアポストロフィによってくくられた拡張MMLを挿入すればよいわけです（見た目で拡張MMLが使用されているのがわかりますね）。

表1を見てください。これらが拡張MMLです。注意したいのは拡張MMLはいままでのMMLと違って横にずら一っと並べて書くことができず，各コマンド間を必ず「，」カンマで区切らなければいけない点です。たとえば，あるMMLトラックをMIDIチャンネル1に送ることにし，音色番号99に切り替えるには，

'T1, P99'

としなくてはいけません。しかし生データなどを送る際には特にコマンドもいらずアポストロフィ「'」の中にデータ数値を書き，同じ要領でそれをカンマで区切るだけでよいので，慣れてしまえばそんなに違和感はなくなります。たとえばMT-32のMIDIチャンネル2のベンドを基準値に直す場合を考えてみますと，

'T2, 225, 0, 64'

というふうになります。表1を見ると「ノートオン」などの命令もありますが，音階のMMLはちゃんと使えるので，これらのお世話になることはほとんどないと思います。

ところで，表1を見て「ちょっと命令が少ないんじゃない？」と感じた方も多くいることでしょう。確かにボリュームコマンドがない，ベンドがない，パンがない，ましてコントロールチェンジもない……。ボリュームについては「ベロシティ」をMMLの「V」でコントロールできますがマスターアウトプットのボリュームがないのはどっちにしろかなりの痛手です。まあ，先ほどの例の「ベンド」にしても3バイトデータをいちいち送らなくてはいけないので，MIDIデータの勉強にはなりますよ。

使用して気づいたことなど

まず，使用して戸惑ったのが「m_assign」命令。おそらく私と同じように「音が鳴りましぇーん」と泣きそうになってしまう人も出てくると思うので，これについて少し説明しておきます。

この命令は周知のとおり，

m_assign（CH, TR）

のように音源チャンネルを「m_alloc」で確保したMMLトラックに割り当てるものです。「OPMDRV2.X」ではCHの部分は前と同じく1～8がFM音源です。MIDI楽器（つまりMIDIチャンネル）は便宜的に9～24となっていますので，たとえばMIDIチャンネル1～8をトラックバッファ番号1～8にアサインするには，

for i=1 to 8

m__alloc (i, 1000)

m__assign (i+8, i)

next

のようにしなくてはいけません。たとえあとで拡張MMLの「T」コマンドを使って初めにアサインしたのとは違うチャンネルを使うとしてもです。よく意味のわからない人はMIDI楽器のときは「i＋8」のようにするんだなと覚えておけばいいでしょう。

MIDI楽器の演奏開始は「md__play」で行いますが「m__play」と多少感覚が違います。というのは「m__play」のときは制御したいチャンネル番号を引数として与えていましたが「md__play」では引数の各ビットがチャンネルに対応しています。具体的には，ビット0～7がFM音源チャンネル1～8に，ビット8～23がMIDIチャンネル1～16に割り当てられており，ビット＝1でそのチャンネルが制御対象になります。具体的な例を示しましょう。たとえばFM音源チャンネル1～8とMIDIチャンネル2と10の演奏を開始したいならば，

```
MD_PLAY(&B0000001000000001011111111)
          ↑                       ↑
       第23ビット              第0ビット
```

のようにします。また「md__stop」や「md__cont」といった演奏制御命令も同様の引数をとります。ですから，

md__play (1, 2, 3, 5, 9, 16)

といったパラメータの与え方はできません。

そのほか気になるのは処理の重さです。OPMDRV.Xと同じ内部処理のまま24トラック化されているようで，MIDIを使うと処理が重くなりデータによってはテンポずれを起こしてしまいます(FM音源だけ使うには困らないのですが)。

そんでもってサンプル

短いサンプルプログラムを用意しました。曲は昔なつかし「ユーフォリー」の「夜の町のテーマ」です。打ち込むのが面倒な人は「OPMDRV2.X」でプログラムするときの流れのようなものをこのサンプルからつかんでください。ちなみに楽器は「MT-32」(CM-32L/CM-64)に対応しています。

「OPMDRV2.X」を常駐させて，「MUSIC2.FNC」(C compiler PRO-68K V2.0のディスクに入っている)を組み込んだBASICを立ち上げ，リストを入力してください。楽器をセットアップして電源を入れたら「RUN」してください。曲が鳴り出すはずです。

MIDI楽器のマニュアルを見れば載っていることですが一応プログラムで使用しているMIDI数値データを解説します。

176＋(MIDIチャンネル－1)，n_1，n_2

はコントロールチェンジです。コントロールn_1に値n_2を書き込んでいます。ちなみに$n_1=7$のときはn_2はボリューム，$n_1=10$のときはn_2はパンポットで，$n_2=0$がもっとも右，$n_2=127$がもっとも左，$n_2=63$が中央です。

224＋(MIDIチャンネル－1)，n_1，n_2

ピッチベンドです。本誌本年度8月号で解説したピッチベンドによるディチューンテクを使っています。ベンドの値は0～16383(2の14乗)の範囲で与えることができます(8192がニュートラル)。いまベンドの値をBとすれば，

n_2＝INT (B/128)

n_2＝B MOD 128

の関係がありサンプルリストではB＝8192やB＝8242などの場合を指定しています。

また，このプログラムを実行するとMT-32のディスプレイにメッセージが出てきますがこれはリスト後半の「mt__print」という関数で表示しています。これは追加命令の「md__wrt」を使って作ったもので，

mt__print (文字変数)

で文字変数の内容をMT-32の画面に出力します。ほとんどお遊びですがエクスクルーシブメッセージの勉強になると思いますので初心者は参考にしてください。

リスト1


```
  10 /*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*
  20 /*                                              /*
  30 /*             A  NIGHT  TOWN                   /*
  40 /*                                              /*
  50 /*               for  MT-32                     /*
  60 /*                                              /*
  70 /*                (C) SACOM                     /*
  80 /*                                              /*
  90 /*          Arranged  by Z.Nishikawa            /*
 100 /*                                              /*
 110 /*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*
 120 m_init():md_init():mt_print("A NIGHT TOWN By Z.N ")
/*メッセージ
 130 /*
 140 str a[256],b[256]
 150 /*
 160 key 2,"md_play()@M":key 12,"md_stop()@M":key 7,"m_trk(":ke
y 19,"save"+chr$(34)+"night
 170 /*
 180 m_tempo(106)
 190 for i=1 to 8:m_alloc(i,1000):m_assign(8+i,i):next
 200 /*                                    ---←注意せよ！！
 210 /*         MELODY
 220 /*
 230 a="r1"
 240 b="L16|: rrbb8a-8b<d-8e->b4&b< |1rrd-d-8>a8a-a-8.g-a-8g-8<
:| rrd-d-8d-8ee-8d->b8.b<cd-4rd-d-ee-4.rre-4re-e-g-e4.r8d-4r d-d
-ee-8d->b8<d-8>b&b2.ra-ab<d-4rd-d-ee-4.rre-4re-eg-a-2a-4a-4g-4e4
e4&eg-ee-&e-4r4>
 250 m_trk(1,"'t2,p19,177,7,127,177,10,63,225,0,64,177,1,40'
 q8 o4 v15"+a)
 260 m_trk(1,"[do]"+b+"[loop]")
 270 /*
 280 m_trk(2,"'t3,p19,178,7,99,178,10,63,226,50,64,178,1,99' r1
6 q8 o4 v10"+a)
 290 m_trk(2,"[do]"+b+"[loop]")
 300 /*
 310 /*         CHORD
 320 /*
 330 a="L16b8.bb4b8.bb4"
 340 b="|:b8.bb4b8.bb8a+8 a8.aa4a8.aa4:| a-8.a-a-4g-8.g-g-4 g-8
.g-g-4a-8.a-a-4 a-8.a-a-4g-8.g-g-4 a-8.a-a-4a-8.a-a-4 a-8.a-a-4g
-8.g-g-4 g-8.g-g-4a-8.a-a-4 a-1a8.a a4a4r4
 350 m_trk(3,"'t4,p06,179,7,100,179,10,63,227,0,64' q6 o4 v13 "
+a)
 360 m_trk(3,"[do]"+b+"[loop]")
 370 /*
 380 a="L16g-8.g-g-4g-8.g-g-4"
 390 b="|:g-8.g-g-4g-8.g-g-8f8 e8.ee4e8.ee4 :| d-8.d-d-4e-8.e-e
-4 c8.cc4e8.ee4 d-8.d-d-4e-8.e-e-4 e8.ee4d8.dd4 d-8.d-d-4e-8.e-e
-4 c8.cc4e8.ee4 d-1e8.ee4e-4r4
 400 m_trk(4,"'t4' q6 o4 v13"+a)
 410 m_trk(4,"[do]"+b+"[loop]")
 420 /*
 430 /*         BASS
 440 /*
 450 a="L16e8.ee8<e8>e8.ee8<e8>"
 460 b="|:e8.ee8<e8>a-8.a-a-8<g8> g-8.g-g-8<g-8>b8.bb8<e-8>:| |
:a8.aa8<a8>b8.bb8<b8> a-8.a-a-8<a-8d-8.d-d-8<d-8>> |1a8.aa8<a8>b
8.bb8<b8> e8.ee8<e8>e8.ee8<e8> :|a1b8.bb8<b8>b8.>b8b<d-e-
 470 m_trk(5,"'t5,p72,180,7,127,180,10,63,228,0,64' q8 o2 v14 "
+a)
 480 m_trk(5,"[do]"+b+"[loop]")
 490 /*
 500 /*         DRUMS
 510 /*
 520 a="L16f+8.f+f+8a+8f+8.f+f+8a+a+
 530 b="|:10f+8.f+f+8a+8f+8.f+f+8a+a+:|f+4a+4a+4a+4f+8.f+f+8a+8
f+4rrf+f+
 540 m_trk(6,"'t10,p50,185,7,127,233,0,64' q8 o2 v8"+a)
 550 m_trk(6,"[do]"+b+"[loop]")
 560 /*
 570 a="L16c8.cc8c8c8.cc8c8
 580 b="|:10c8.cc8c8c8.cc8c8:|c4c4c4c4c8.cc8d8c4rr<c>f
 590 m_trk(7,"'t10' q8 o2 v10"+a)
 600 m_trk(7,"[do]"+b+"[loop]")
 610 md_play(&HFFFF00):end
 620 func mt_print(m;str)
 630 dim char d(7)={&HF0,&H41,&H10,&H16,&H12,&H20,&H0,&H0}
/*command and address data
 640 int i,s
 650 char e,a,l
 660 if len(m)>20 then print"メッセージが長すきるたよ…。こんな
のを送るなんてオラには出来ねぇ":return()          /* ERROR
 670 for i=0 to 7:md_wrt(d(i)):next
 680 s=0:l=len(m)
 690 for i=1 to l
 700 a=asc(mid$(m,i,1))
 710 md_wrt(a)
 720 s=s+a
 730 next
 740 for i=5 to 7
 750 s=s+d(i)
 760 next
 770 e=128-(s and &HFF):e=e and 127
 780 md_wrt(e)          /*send checksum data
 790 md_wrt(&HF7)       /*end of exclusive
 800 endfunc
```

BASICコンパイラ派に贈る便利ソフト

XBAStoC CHECKER

Nishikawa Zenji 西川 善司

X-BASICでプログラムを作成しXC（またはGCC）でコンパイル。これが意外と効率のよい開発方法なのです。ここではBASICユーザーを支援する強力なツールを紹介します。これまでコンパイルできなかったプログラムもこれで大丈夫？

いまどき，C言語が使えないパソコンはあまりない。では，どこで選ぶか？

お答えします。それは，BASICもコンパイルできる「C compiler PRO-68K」が動くX68000です。

目のつけどころがシャープでしょ。

と，いうわけで若葉マークのX68000ユーザーのなかにはこの事実を知らなかった人もいるんじゃないかな。そう，「C compiler PRO-68K」（以下XC）はX-BASICのプログラムをコンパイルすることができるんです。

具体的にはまずBASICのソースリストをCのソースに変換し，これをコンパイルし機械語の実行ファイルを作成します。これは大変便利な機能なんだけれど，BASICならなんの問題もなしに動いていたものが，いざコンパイルしてみると動作がちょっと変，なんてことがよくあるのです。

今回発売となった「XBAStoC CHECKER」（以下チェッカ）はこういったX-BASIC→Cの変換段階でプログラマの目的と違って変換されてしまう部分をわかりやすく指摘してくれるソフトなのです。

チェッカの概要

さて，実際にこのソフトは大きく分けて2つの動作モードがあります。

X68000用　5"2HD版　9,800円（税別）
シャープ　☎03(260)1161

写真1　起動時の画面

ひとつはグラフィカルモード。これは，すべての操作をビジュアルに行えるもので画面写真のようなメニュー画面をマウスで次々にクリックしていくことにより作業が進められます。こちらのモードで動作させるにはG-RAM（グラフィックRAM）をRAMディスクとして使用していないことが大前提です。理由は見てのとおり，メニューをグラフィックで描いているためです。

もうひとつは一般的なコマンドラインで動作させるモードです。まあ，通常の外部コマンドのように動作するモード，と考えていただければ結構でしょう。ですから，バッチファイルなどを使った自動処理にはこちらが向いていますね。

動作環境における注意がいくつかあるのでこれを挙げておきます。まず，両モードとも動作にはテンポラリディスク（まあ，コンピュータがメモ帳代わりにディスクを使うと考えてください）が必要です。チェッカは動作にあたっていくつかの中間ファイルを作成しますので，ある程度フリーエリアのあるディスク上で作業するか，または，

A>TEMP ?:

を実行して十分空きのあるディスク（RAMディスク/ハードディスクでも可）をテンポラリとして指定する必要があります。

また，両モードとも浮動小数点演算パッケージが，またグラフィカルモードではFM音源ドライバをデバイスドライバとして登録しておく必要があります。

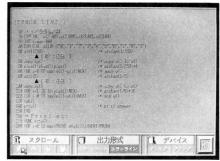



写真2　エラーレポート

写真3　ここがエラー

作業手順はこんな感じ

さて，実際にリストをチェッカにかけてみることにしましょう。ここでは，グラフィカルモードを使用して話を進めます。

立ち上げるとタイトルのあと，写真1のようなファイル選択画面となります。ここでチェックしたいプログラムをマウスで選択します。ちなみに拡張子が「.BAS」以外のファイルは表示されませんのであしからず。

さて，画面右上の「エラーレポート」と「エラーライン」ですが，これはチェッカがユーザーにどんな情報を提供すればいいかの指定をするものです。「エラーレポート」ではエラーの概要とその行番号を表示してくれます。「エラーライン」では実際にソースリストのどの部分にトラブルがあるのかを指摘してくれます。まあ，通常は両方のスイッチをオンにしておくといいでしょうね。

その下の「デバイス」というのは先ほどスイッチ指定した「エラー情報」をどこへ出力するかを決定するものです。「画面」と「プリンタ」は読んで字のごとくですが，「ディスク」では「エラー情報」をファイルに書き出してくれます。

さあ，あとは「実行」で実際にチェックしてみましょう。

チェック終了メッセージのあと，初めの出力形式の設定のところで「エラーレポート」をオンにしてあれば写真2のような画面が表示されることでしょう。「エラーライン」のみをオンにしてあると写真3のような画面が出てきます。また，両方のスイッチをオンにしてあれば画面下の「出力形式」のメニューをマウスでクリックすることによって2つの「エラー情報」を切り替えることができます。

この時点で「デバイス」を変更し「エラー情報」をディスクやプリンタに出力することも可能なので，

・初めは「デバイス」を「画面」に設定しておき

・必要ならば「ディスク」や「プリンタ」に出力し保存する

というのが一般的な使い方でしょうか。

さあ，エラーを確認したあとは，BASICやエディタに帰り，サクサクとデバッグをすればいいわけです。

はふ，謎が解けた！

このソフトのマニュアルの後半にはチェッカの出力する「エラー情報」の詳細とその対応策が載っていますが，これは大変参考になりますぞ。ここには「原因」という項目があるのですが，これを読めば「XC」がX-BASICのプログラムをどう変換してしまうのか，また，C言語とBASICの相違点などを知ることができるのです。

そうですね，たとえば，ゲームなどのキー入力の処理なんかで，

X＝X＋(A＝"4")－(A＝"6")

ということをしますが，BASICでは正常に動作しても，これをCに変換した場合には不都合が生じます。これは，BASICとC言語とでは論理演算の出力結果が違うために起こるものです。具体的には，

BASIC	真：	－1
	偽：	0
C	真：	1
	偽：	0

です。ですから上の例をコンパイルするときには「＋」と「－」を入れ替える必要があるでしょう。

このように長い間原因不明だったバグの正体が，ここを読むにつれてあれよあれよと解明されていく気分はさながら推理小説の後半を読んでいるようですよ。

まあ，これは私ひとりの願望かもしれませんが，わかりきった「エラー」もしくは「注意事項」は自動的に直してくれる機能がほしかったですね。たとえば文字列の最後を，

A＝"ABCDEFGHIJKLMN

のように「"」でくくっていないとチェッカは「注意」を促してくるのですが，自動的に「"」をつけ足してくれるくらいの気の利いた処理をしてもバチは当たらないと思うのですが。まあ，勝手にやられると気持ち悪いという人もいるでしょうからスイッチ指定できるようにするとか，ね。

まとめ，とその他気づいた点

ゲーム誌「LOGIN」なんかの「ソフコン」の入賞ソフトにはX68000用のものが比較的多いですが，その中にはX-BASICからコンパイルしたものがよく見受けられます(「C ON Z」とかありましたね)。また，多くの同人ソフトのグループなんかもこの「XC」のBASICコンパイル機能を愛用しているようですね。まさに，今回の「XBAStoC CHECKER」はこういう人たちにおすすめです。しかし，こんな便利なもの，「XC」の本体につければよかったのにねぇ。

最後に気づいた点をいくつか。

グラフィカルモードは大変便利ですが，G-RAMを使用しているのはちと痛いですね。というのは，RAMを何Mバイトも増設している人はともかく，多くの人はG-RAMをテンポラリディスクにあてていると思うのです。それなのに，G-RAMをRAMディスクにしていると使用不可能というのは痛い。だいたい，あんな派手な画面にする必要はないでしょうに。「XC v. 2.0」のソースコードデバッガや「COMMAND.X」用外部コマンドの「FORMAT.X (v. 2.00以降)」「SWITCH.X (v. 2.00以降)」を見てもわかるようにテキスト画面だけでも十分見やすい画面は作れるのです。

いずれにせよ，大変便利なことは確か。まだ，機械語はわからないけどゲームなんかを作ってみたいな，なんて思っている人は「XC」とセットで買うといいかもよ。

図　マニュアルはいろいろ参考になる

28．　注意「int関数は実行結果が異なる可能性がある」

原　因

int関数に渡されたパラメータ（数値・変数）がマイナス値の場合、実行結果がおかしくなることがあります。
例えば、"print int(-3.3#)"を実行した場合、インタプリタでは"-4"、コンパイラでは"-3"という結果が返ってきます。

解決策

「修正例」で示すように、最大整数値（int）の算出はユーザー関数の方で行うようにします。
標準関数"int()"は、なるべく使用しないようにします。

問題例

```
 100 print int(-2.8#)
              ▲ ( 28 :注意 )
```

修正例

次のようにユーザー側で新しいint関数を作成し（"int2()"）、使用するようにします。インタプリタのときでも、"int()"はなるべく使わないようにしてください。

```
  100 print int2(-2.8#)
  110 end
10000 func int  int2(a;float)    /* ユーザーint関数
10010   float b
10020   b=fix(a)
10030   if a<b then b=b-1
10040   return(b)
10050 endfunc
```

THE SENTINEL

●STACKコンパイラ登場

インタプリタ言語STACKにコンパイラ版ができました。STACKとフルコンパチですので，インタプリタで開発，テストラン，そしてコンパイルして実行と，より手軽にアセンブラによらない高速アプリケーション開発が扱えます。

作者はもうお馴染みの平井真二氏です。

もともと，S-OSオリジナル言語のなかでも，一風変わった風貌を持つインタプリタとコンパイラシステム。スタック型という考え方に慣れさえすれば，常用の言語としてアセンブラを使うでもない分野（つまりほとんどの処理）をこなすことができるでしょう。実用面でももちろんマルの性能です。

手軽さのなかには，全体の小ささとあいまって「コンパイルが高速」だということも含まれます。インタプリタとあわせて使用するのですが，コンパイルが高速なら，コンパイラの存在を感じさせないシステムを組み上げることも可能。

S-OS用のオリジナル言語というとやはりSLANGの優秀さが目につきますが，インタプリタ/コンパイラの環境はやはり捨てがたいものがあります。SLANGインタプリタなんてできないかな……。

第101部 STACKコンパイラ

●Cは延期

予告していたC言語の移植はちょっと延期となりました。期待していた方ごめんなさい。

豊富なラインアップを誇るS-OSの言語処理系ですが，なぜか欠けていたのがC言語でした。S-OSスタート当時は「噂のC言語を使ってみたい」という要望が多かったものです。伝説の処理系だったC言語も，いまやアセンブラ代わりの開発言語として日常的に定着した観があります。

これまでにも何度かC言語をサポートしようという気運が盛り上がったことはあったのですが，いずれも実現しなかったのは「処理系だけなら作りますよ。ライブラリのほうはお願いしますね」というスタッフの弁に代表されるように，処理系作りの難しさよりライブラリを揃えることの面倒臭さが原因となっていたのではないでしょうか。

C言語を完全にアセンブラ代わりに使うのならともかく，高級言語的に使用するのならライブラリが充実していなければ手も足も出ません。C言語を使いたいと要望していた皆さんのなかには，アセンブラは無理でもCならS-OS用のアプリケーションを作れそうだという方が多く，ライブラリなしのC言語など考えられなかったのです。

ですから，CP/M上のC言語を持ってこようというのは，むしろ自然な発想といえるでしょう。

ただし，誌面に掲載できるのは変更箇所と手順に限られますから，実際にコンパイラのインプリメントを行うのは非常に面倒な作業となりそうです。少なくとも，CP/M，MACRO-80ほか，これまで準備してきたCP/Mファイルコンバータ，やWZD，WLK，WLBなどのシリーズがすべて必要となります。覚悟だけはしておいてください。

●S-OSの系譜（16）

マシン語ファイルの共通化を果たしたS-OSと，グラフィックデータの共通化を果たした高機能グラフィックパッケージMAGIC。両者の能力を融合するとどんなものができるのか。この試みは1987年3月号で行われました。掲載されたプログラムはMAGE（メイジ）と命名されました。MAGEとは魔法使いの意。MAGIC（魔法）を使いこなすMAGEの登場です。

MAGEはピクチャーエディタとストーリーエディタの2つのプログラムから成っています。ピクチャーエディタで線画の絵を描いておき，それをパラパラ漫画のように順次表示することによってアニメーションしようというのです。作成された絵はMAGICのデータですからMAGICが移植されたMZ-1500/2000/2200/2500，X1/turbo，PC-8801シリーズ，そしてSMC-777で共通に扱うことができます。もちろんMAGE自身はS-OSのアプリケーションですからこれら全機種で共通のプログラム。グラフィックを使ったアプリケーションでプログラムもグラフィックデータも共通という大きなイベントが繰り広げられたのです。

DōGAプロジェクトの出現したいまとなっては，思わず「線画？」と問いかけたくなるかもしれませんが，64Kバイトのメモリではフルアニメーションなど無理なことと切り捨てた結果，線しか表現できないことがかえってカリグラフのような独特の魅力をもたらしていたものです。ピクチャーエディタで描いた絵はストーリーエディタによって管理され，タイムテーブルに従って順次表示されていきます。いかに少ない絵で効果的に見せるか，手腕の問われるところでした。

MAGICは続く1987年4月号でMZ-80B/B2にも移植されました。MZ-80B/B2のグラフィックは320×200。しかもG-RAMは2プレーンしかありませんが，解像度の違いは内部で640ドット→320ドットの変換を行ってデータの共通化を実現しています。もちろんMAGEにも対応。MZ-80B/B2ユーザーからのこの投稿プログラムはユーザーの底力といったものを見せつけてくれました。

全機種共通S-OS“SWORD”要

STACKコンパイラ

FORTHとBASICを足したようなスタック型言語STACKがコンパイラになりました。インタプリタ上のプログラムがこれまで以上の速度で実行できます。使用の際にはSTACKインタプリタが必要です。

Hirai Shinji
平井 真二

STACKコンパイラはSTACKフルコンパチのコンパイラです。ランタイムルーチンを含めても4Kバイト弱とコンパクトですが，これはSTACKのセミコンパイル結果を利用しているからです。そのため本コンパイラを利用するにはSTACKが必要です。

速度が気になると思いますが，STACKのセミコンパイラと比べると1.5～2.5倍ほど速くなります。

入力&実行方法

まず，リスト1のダンプリストをMACINTO-Cなどのマシン語入力ツールから打ち込んで，実行アドレス$3A13_H$でセーブしてください。

使い方は，まずコンパイルしたいプログラムをSTACKのCコマンドでセミコンパイルしてください。次にS-OSのモニタに戻り，STACKコンパイラをロードして，

　　#J3A13

と入力して起動してください。

コンパイラは次のような質問をしてくるので，16進4桁で答えてください。

TEXT ADDRESS　：

　　セミコンパイル結果の格納先頭番地。

OBJECT ADDRESS：

　　オブジェクトの発生開始番地（セミコンパイル結果と重ねることはできない）。

VARIABLE TOP　：

　　変数領域の先頭番地。

STACK TOP　：

　　パラメータスタックの先頭番地。

RET_STACK TOP　：

　　リターンスタックの先頭番地。

OFFSET ADDRESS：

　　オフセット。通常は0000とします。

以上でコンパイルが開始されます。しばらくすると，オブジェクトの終了アドレスを表示してS-OSに戻ります。なお，このアドレスはオフセットを含めた値なので注意してください。

コンパイルが終了したら，ランタイムルーチン（3000_H～$3A12_H$）とオブジェクトを含めた範囲で実行アドレスをオブジェクトの先頭番地としてセーブしてください。

　　#Jオブジェクトの先頭番地

でコンパイルしたプログラムが動きます。

サンプルプログラム

STACKコンパイラの動作をチェックするためのサンプルプログラムを掲載します。エラトステネスのふるいを用いた素数を求めるプログラムです。

まず，STACKインタプリタのエディタで入力し，

　　]C6000

でセミコンパイルしてください。

次にモニタに戻り，STACKコンパイラを起動し，

　　TEXT ADDRESS　：6000
　　OBJECT ADDRESS：4000
　　VARIABLE TOP　：4200
　　STACK TOP　：5000
　　RET_STACK TOP　：5800
　　OFFSET ADDRESS：0000

と入力すると，OBJECT ENDのアドレスを表示して終了します。実行はS-OSのモニタからJ4000です。

サンプル

```
      1 ; ｴﾗｽﾄﾃﾈｽ ﾉ ﾌﾙｲ
      2 8190  .M
      3 $8000 .F
      4 ;
      5 #F #M 0 FILL ; CLEAR
      6 2 PRINT
      7 3 .I
      8 ;
      9 %1
     10  #M #I < IF 1 BELL END
     11  #F #I + PEEKB =0 IF #I COPY
PRINT GOSUB 2
     12  INC I  INC I
     13  GOTO 1
     14 ;
     15 %2
     16  .B
     17  2 #M #B / DO
     18  1 #F #B I? * +  POKEB
     19  LOOP! RET
```

最　後　に

コンパイルはかなり安易な方法で行ってますが，文法解析およびエラーチェックがセミコンパイル時に終わっているためコンパイル速度はそれほど遅くはありません。

ランタイムルーチンは3000_H～$3A12_H$に固定されてますがそれほど問題はないでしょう。セミコンパイラと比べ，それほど速くはなりませんが，実行時にインタプリタがいらない，512以上のラベルを使っても速度が落ちない，マシン語サブルーチン化ができるなどのメリットがあります。普通に使う分にはセミコンパイラで十分だと思いますが，もっとスピードがほしいというときにはぜひ本コンパイラを使ってみてください。6月号のSQUASH！も見違えるように操作性がよくなります。

今後は，またまた時代の流れに逆らって2～3Kバイト程度の記号型言語でも作ろうと思っています。

リスト1　STACKコンパイラ

```
3000 00 00 00 AE 00 00 00 00 : AE
3008 00 00 00 00 00 32 08 30 : 6A
3010 ED 53 09 30 22 0B 30 C9 : 9F
3018 DD E1 D1 E1 19 E5 DD E9 : 34
3020 DD E1 D1 E1 B7 ED 52 E5 : 4B
3028 DD E9 DD E1 D1 E1 CD B0 : B3
3030 30 E5 DD E9 DD E1 D1 E1 : 4B
3038 CD C3 30 E5 DD E9 DD E1 : 29

3040 D1 E1 CD C3 30 D5 DD E9 : 0D
3048 DD E1 D1 E1 CD C3 30 E5 : 15
3050 D5 DD E9 DD E1 E1 D1 B7 : C2
3058 ED 52 21 00 00 20 01 23 : A4
3060 E5 DD E9 DD E1 D1 E1 B7 : D2
3068 ED 52 21 01 00 38 01 2B : C5
3070 E5 DD E9 DD E1 E1 D1 18 : 33
3078 EE DD E1 E1 D1 B7 ED 52 : 54

--------------------------------
SUM: 96 80 11 6C EE F4 61 2D 2596

3080 21 01 00 20 01 2B E5 DD : 30
3088 E9 DD E1 D1 E1 7D A3 6F : E8
3090 7C A2 67 E5 DD E9 DD E1 : EE
3098 D1 E1 7D B3 6F 7C B2 67 : E6
30A0 E5 DD E9 DD E1 D1 E1 7D : 98
```

```
30A8 AB 6F 7C AA 67 E5 DD E9 : 52
30B0 4D 44 3E 10 21 00 00 29 : 29
30B8 CB 23 CB 12 30 01 09 3D : 42
30C0 20 F5 C9 4B 42 5D 54 3E : 5A
30C8 10 21 00 00 CB 23 CB 12 : FC
30D0 ED 6A E5 B7 ED 42 E1 38 : 3B
30D8 03 ED 42 13 3D 20 ED EB : 7A
30E0 C9 DD E1 D1 E1 D5 E5 DD : D0
30E8 E9 DD E1 E1 D1 C1 D5 E5 : D4
30F0 C5 DD E9 DD E1 CD CA 1F : FF
30F8 6F 26 00 E5 DD E9 DD E1 : FE
----------------------------------
SUM: 05 3E CE BB 6E F2 2C 95 3D66

3100 CD D0 1F 6F 26 00 E5 DD : 13
3108 E9 DD E1 CD 21 20 6F 26 : 4A
3110 00 E5 DD E9 DD E1 2A 06 : 99
3118 30 54 5D 19 19 7D 84 67 : 7B
3120 85 6F 11 54 00 19 22 06 : 9A
3128 30 E5 DD E9 DD E1 D1 E1 : 4B
3130 63 CD 1B 20 6F 26 00 E5 : E5
3138 DD E9 DD E1 E1 7D CD C1 : 70
3140 1F DD E9 DD E1 E1 CD BE : 0F
3148 1F DD E9 DD E1 E1 CD 73 : C4
3150 38 DD E9 DD E1 E1 7D CD : E7
3158 F4 1F DD E9 DD E1 E1 7E : F6
3160 FE 22 28 0B FE 0D CA 6F : 97
3168 31 CD F4 1F 23 18 F0 DD : 19
3170 E9 DD E1 E1 7E FE 22 28 : 4E
3178 F6 FE 0D CA 6F 31 0E 1C : 95
----------------------------------
SUM: 53 70 C2 D1 F8 F3 A4 09 7DFA

3180 FE 52 28 16 0C FE 4C 28 : 0C
3188 11 0C FE 55 28 0C 0C FE : AE
3190 44 28 07 0E 0C FE 43 28 : F6
3198 01 0C 79 CD F4 1F 23 18 : A1
31A0 D3 3E 0D C3 F4 1F DD E1 : B2
31A8 E1 7D CD 30 20 DD E9 DD : 1E
31B0 E1 E1 45 78 B7 28 05 CD : 30
31B8 C4 1F 10 FB DD E9 DD E1 : 72
31C0 D1 E1 63 CD 1E 20 DD E9 : E6
31C8 C1 2A 02 30 2B 70 2B 71 : 54
31D0 22 02 30 EB E9 2A 02 30 : 84
31D8 5E 23 56 23 22 02 30 EB : 39
31E0 E9 D1 2A 02 30 2B 72 2B : DE
31E8 73 22 02 30 EB E9 DD E1 : 59
31F0 E1 7C B5 28 0A 2A 02 30 : A0
31F8 23 23 22 02 30 DD E9 2A : 8A
----------------------------------
SUM: 1F 0F C3 13 85 0B DA AD 279A

3200 02 30 5E 23 56 EB E9 DD : BA
3208 E1 E1 D1 ED 73 04 30 ED : 14
3210 7B 02 30 DD E5 E5 D5 ED : 16
3218 73 02 30 ED 7B 04 30 DD : 1E
3220 E9 DD E1 ED 73 04 30 ED : 28
3228 7B 02 30 D1 E1 FD E1 13 : 50
3230 B7 ED 52 38 13 19 FD E5 : 3C
3238 E5 D5 ED 73 02 30 ED 7B : B4
3240 04 30 FD E5 DD E1 DD E9 : 9A
3248 ED 73 02 30 ED 7B 04 30 : 2E
3250 DD E9 ED 7B 00 30 3A 08 : A0
3258 30 ED 5B 09 30 2A 0B 30 : 16
3260 C9 DD E1 11 00 00 2A 02 : C4
3268 30 19 5E 23 56 D5 DD E9 : BB
3270 DD E1 11 06 00 18 EF DD : B9
3278 E1 E1 ED 73 04 30 ED 7B : BE
----------------------------------
SUM: 86 E7 63 89 E6 F5 22 88 CE8D

3280 02 30 E5 ED 73 02 30 ED : 96
3288 7B 04 30 DD E9 DD E1 2A : 5D
3290 02 30 01 00 AE B7 ED 42 : C7
3298 ED 73 04 30 ED 7B 02 30 : 2E
32A0 E1 ED 73 02 30 ED 7B 04 : DF
32A8 30 E5 DD E9 DD E1 2A 02 : C5
32B0 30 5E 23 56 23 73 23 72 : 32
32B8 DD E9 DD E1 C1 2A 0B 30 : AA
32C0 ED 5B 09 30 3A 08 30 ED : E0
32C8 43 CC 32 CD 00 00 22 0B : 3B
32D0 30 ED 53 09 30 32 08 30 : 13
32D8 DD E9 DD E1 E1 7D 32 08 : 1C
32E0 30 DD E9 DD E1 3A 08 30 : 26
32E8 6F 26 00 E5 DD E9 DD E1 : FE
32F0 D1 ED 53 09 30 DD E9 DD : ED
32F8 E1 ED 5B 09 30 D5 DD E9 : FD
----------------------------------
SUM: 18 CA 6C D7 51 08 0A 38 2A55

3300 DD E1 E1 22 0B 30 DD E9 : C2
3308 DD E1 2A 0B 30 E5 DD E9 : CE
3310 DD E1 E1 5E 16 00 D5 DD : C5
3318 E9 DD E1 E1 5E 23 56 D5 : 34
3320 DD E9 DD E1 E1 D1 73 DD : 86
3328 E9 DD E1 E1 D1 73 23 72 : 61
3330 DD E9 DD E1 C1 ED 58 16 : A0
3338 00 D5 DD E9 DD E1 E1 C1 : FB
3340 ED 69 DD E9 DD E1 E1 6C : 27
3348 26 00 E5 DD E9 DD E1 E1 : 70
3350 26 00 E5 DD E9 DD E1 E1 : 70
3358 7D 6C 67 E5 DD E9 DD E1 : B9
3360 E1 7D 2F 6F 7C 2F 67 E5 : F3
3368 DD E9 DD E1 C1 E1 41 CB : 32
3370 3C CB 1D 10 FA E5 DD E9 : D9
3378 DD E1 C1 E1 41 29 10 FD : D7
----------------------------------
SUM: B0 EB 3D C1 03 EC C9 4F 1A5A

3380 E5 DD E9 DD E1 CD 18 20 : 6E
3388 26 00 E5 DD E9 DD E1 CD : 5C
3390 18 20 6C 26 00 E5 DD E9 : 75
3398 DD E1 E1 CD A1 33 E5 DD : 02
33A0 E9 7D 2F 6F 7C 2F 67 23 : 39
33A8 C9 7B 2F 5F 7A 2F 57 7D : 4F
33B0 2F 6F 7C 2F 67 01 01 00 : B2
33B8 EB 09 EB 0B ED 4A C9 DD : C7
33C0 E1 E1 D1 1A 77 FE 0D 28 : 57
33C8 0A FE 22 28 04 23 13 18 : A4
33D0 F2 36 0D DD E9 DD E1 C1 : 7A
33D8 E1 D1 1A 77 FE 0D 28 0D : 83
33E0 FE 22 28 07 23 13 0B 78 : 08
33E8 B1 20 EF 36 0D DD E9 DD : A6
33F0 E1 C1 FD E1 E1 5D 54 7E : 90
33F8 FE 22 28 07 FE 0D 28 03 : 85
----------------------------------
SUM: 18 59 36 70 26 D0 DC 14 F1D7

3400 23 18 F4 2B E5 B7 ED 52 : 35
3408 E1 28 05 0B 78 B1 20 F3 : 55
3410 FD E5 D1 EB 18 AD DD E1 : 21
3418 D9 C1 D9 C1 E1 D1 0B 78 : 69
3420 B1 28 08 1A FE 22 FE 0D : 26
3428 13 18 F3 D9 C5 D9 C1 1A : 70
3430 77 FE 22 28 0B FE 0D 28 : FD
3438 07 23 13 0B 78 B1 20 EF : 80
3440 36 0D DD E9 DD E1 D1 E1 : 79
3448 7E FE 0D CA C3 33 FE 22 : 69
3450 CA C3 33 23 18 F2 DD E1 : AB
3458 E1 01 00 00 7E FE 0D 28 : 93
3460 08 FE 22 28 04 23 03 18 : 92
3468 F3 C5 DD E9 DD E1 C1 E1 : DE
3470 11 01 00 7E B9 28 0F FE : 7E
3478 0D 28 08 FE 22 28 04 13 : 9C
----------------------------------
SUM: 94 02 F7 6B 8E E8 71 F2 312C

3480 23 18 F0 11 00 00 D5 DD : EE
3488 E9 DD E1 E1 D1 1A FE 0D : 7E
3490 28 0C FE 22 28 08 46 90 : 5A
3498 20 13 23 13 18 EF 01 00 : 71
34A0 00 7E FE 0D 28 0F FE 22 : E0
34A8 28 0B 0B 18 08 01 01 00 : 60
34B0 30 03 01 FF FF C5 DD E9 : BD
34B8 DD E1 C1 D1 E1 09 EB C1 : E6
34C0 ED 4A E5 D5 DD E9 DD E1 : 75
34C8 D1 C1 E1 B7 ED 52 EB E1 : 35
34D0 ED 42 E5 D5 DD E9 DD E1 : 6D
34D8 D1 E1 D9 FD 21 00 00 21 : CA
34E0 00 00 D1 C1 D9 06 20 CB : 5C
34E8 3C CB 1D CB 1A CB 1B D9 : C8
34F0 30 04 FD 19 ED 4A CB 23 : 6F
34F8 CB 12 CB 11 CB 10 D9 10 : 7D
----------------------------------
SUM: 3C 90 F7 30 94 3E 65 E1 9BED

3500 E6 D9 E5 FD E5 DD E9 DD : 29
3508 E1 CD 2C 35 D9 C5 D9 C5 : 4B
3510 DD E9 DD E1 CD 2C 35 D9 : 8B
3518 E5 D9 E5 DD E9 DD E1 CD : F4
3520 2C 35 D9 E5 D9 E5 D9 C5 : 7B
3528 D9 C5 DD E9 FD E1 D1 21 : 34
3530 00 00 D9 D1 21 00 00 D9 : A4
3538 C1 D9 C1 D9 3E 20 F5 CB : 52
3540 21 CB 10 D9 CB 11 CB 10 : 8C
3548 D9 ED 6A D9 ED 6A D9 B7 : F0
3550 ED 52 D9 ED 52 D9 38 0E : 76
3558 03 78 B1 20 03 D9 03 D9 : 04
3560 F1 3D 20 DA FD E9 19 D9 : 00
3568 ED 5A D9 18 F3 DD E1 C1 : AA
3570 D1 E1 CD 7C 35 D9 D5 D9 : B7
3578 E5 D5 DD E9 F5 C5 D9 C1 : D4
----------------------------------
SUM: CD 0A CA 7E D0 22 FE B4 366B

3580 21 00 00 11 00 00 D9 3E : 49
3588 20 EB 29 EB ED 6A D9 EB : 3A
3590 ED 6A EB ED 6A D5 E5 CD : 20
3598 AA 35 E1 D1 38 03 CD AA : 43
35A0 35 D9 38 01 1C 3D 20 E1 : A1
35A8 F1 C9 EB B7 ED 42 EB D0 : 46
35B0 67 7D D6 01 6F 7C 26 00 : CC
35B8 C9 DD E1 C1 D1 21 00 00 : 3A
35C0 D9 11 00 00 21 00 00 D9 : E4
35C8 3E 10 CB 38 CB 19 30 05 : 6A
35D0 19 D9 ED 5A D9 CB 23 CB : CB
35D8 12 D9 CB 13 CB 12 D9 3D : BC
35E0 20 E8 D9 E5 D9 E5 DD E9 : 4A
35E8 DD E1 C1 D1 21 00 00 79 : EA
35F0 B7 28 0C CB 3F 30 01 19 : 3F
35F8 CB 23 CB 12 C3 F0 35 E5 : 98
----------------------------------
SUM: EF 6D C3 6C 64 59 D4 97 FBF9

3600 DD E9 DD E1 C1 D1 E1 B7 : AE
3608 ED 42 4D 44 E1 ED 52 38 : 18
3610 09 7C B5 B0 B1 20 08 E5 : A8
3618 DD E9 21 FF FF 18 F8 21 : 16
3620 01 00 18 F3 DD E1 E1 11 : BC
3628 00 00 CB 7C 28 01 1B D5 : 60
3630 E5 DD E9 DD E1 E1 11 00 : 5B
3638 00 7E 23 FE 0D 28 08 FE : DA
3640 22 28 04 53 5F 18 F2 D5 : DF
3648 DD E9 DD E1 D1 E1 01 01 : 38
3650 00 CB 7A 20 0D CB 7C 20 : D9
3658 06 B7 ED 52 38 01 0B C5 : 05
3660 DD E9 CB 7C 28 F8 18 F1 : 36
3668 DD E1 E1 11 01 00 7C B5 : E2
3670 28 01 1B D5 DD E9 DD E1 : 9D
3678 E1 23 E5 DD E9 DD E1 E1 : 4E
----------------------------------
SUM: 5E 6C E3 03 A9 64 14 FC F554

3680 2B E5 DD E9 DD E1 D1 21 : 86
3688 00 00 CD A0 37 11 C4 37 : B0
3690 CD E8 1F DD E9 DD E1 D1 : 29
3698 E1 18 EF DD E1 E1 11 00 : 98
36A0 00 CB 7C 28 08 3E 2D CD : AF
36A8 F4 1F CD A1 33 EB 18 DA : 91
36B0 DD E1 D1 E1 CB 7C 28 D2 : B1
36B8 3E 2D CD F4 1F CD A9 33 : F4
36C0 C3 8A 36 DD E1 E1 D1 E5 : D8
36C8 21 00 00 CD A0 37 E1 11 : B7
36D0 C4 37 C3 C3 33 DD E1 C1 : 33
36D8 D1 E1 C5 18 EE DD E1 D1 : 0C
36E0 E1 CD BE 1F EB CD BE 1F : 20
36E8 DD E9 DD E1 FD E1 FD 7E : DD
36F0 00 FE 2D 28 06 CD D0 37 : 2D
36F8 E5 DD E9 FD 23 CD D0 37 : 9F
----------------------------------
SUM: 04 10 0E 8B B6 3C 6C 68 C928

3700 CD A1 33 18 F3 DD E1 FD : 67
3708 E1 CD EE 37 E5 D5 DD E9 : 53
3710 DD E1 FD E1 CD 87 38 E5 : 0D
3718 DD E9 DD E1 CD 18 20 26 : AF
3720 00 ED 5B 76 1F CD D3 1F : 9C
3728 1A FE 1B 28 0C 19 EB E1 : 4C
3730 1A B7 28 06 77 13 23 18 : C4
3738 F7 E1 36 0D DD E9 DD E1 : 9F
3740 C1 D1 E1 ED B0 DD E9 DD : B3
3748 E1 C1 D1 E1 ED B8 DD E9 : BF
3750 DD E1 D1 C1 E1 0B 73 54 : 03
3758 5D 13 ED B0 DD E9 DD E1 : 91
3760 D1 E1 E5 D5 E5 D5 DD E9 : EC
3768 DD E1 E1 E1 DD E9 DD E1 : 04
3770 D1 E1 D9 D1 E1 D9 E5 D5 : D0
3778 D9 E5 D5 DD E9 CD CD 1F : 12
----------------------------------
SUM: C7 C9 B3 65 D8 20 56 A3 84C1

3780 C0 C3 52 32 C3 D9 1F C3 : 85
3788 D6 1F DD E1 E1 CD 94 1F : 14
3790 6F 26 00 E5 DD E9 DD E1 : FE
3798 C1 E1 79 CD 9A 1F DD E9 : 67
37A0 01 00 00 C5 01 0A 00 CD : 9E
37A8 7C 35 D9 3E 30 83 D9 C1 : 15
37B0 03 F5 7C B5 B2 B3 20 EB : 99
37B8 41 21 C4 37 F1 77 23 10 : F8
37C0 FB 36 0D C9 00 00 00 00 : 07
37C8 00 00 00 00 00 00 00 00 : 00
37D0 21 00 00 FD 7E 00 FE 30 : CA
37D8 D8 FE 3A D0 FD 23 29 54 : 7D
37E0 5D 29 29 19 D6 30 85 30 : 83
37E8 01 24 6F C3 D3 37 21 00 : 82
37F0 00 11 00 00 FD 7E 00 FE : 8A
37F8 30 D8 FE 3A D0 FD 23 D6 : 06
----------------------------------
SUM: 09 9E 9E 60 E0 6A 79 BD BD6E

3800 30 EB 29 EB ED 6A 44 4D : 17
3808 C5 42 4B EB 29 EB ED 6A : A8
3810 EB 29 EB ED 6A EB 09 EB : 35
3818 C1 ED 4A 06 00 4F EB 09 : 41
3820 EB 0E 00 ED 4A 18 CD DD : F2
3828 E5 DD 21 64 38 FD 21 6E : 0B
3830 38 3E 04 06 05 0E 2F DD : 9F
3838 5E 00 DD 56 01 0C B7 ED : 42
3840 52 30 FA 19 B7 C4 56 38 : 9E
3848 FD 71 00 DD 23 DD 23 FD : 6B
3850 23 10 E2 DD E1 C9 3D 08 : E1
3858 3E 30 B9 20 04 08 0E 20 : 81
3860 C9 08 AF C9 10 27 E8 03 : 6B
3868 64 00 0A 00 01 00 00 00 : 6F
3870 00 00 00 CD 27 38 FD 21 : 4A
3878 6E 38 06 05 FD 7E 00 CD : F9
----------------------------------
SUM: 52 8D FF 04 FC 0D A2 0E E80B

3880 F4 1F FD 23 10 F6 C9 21 : 23
3888 00 00 DD 7E 00 FE 30 D8 : 61
3890 FE 47 D0 FE 3A 38 05 FE : 88
3898 41 D8 D6 07 D6 30 29 29 : 4E
38A0 29 29 85 30 01 24 6F DD : 78
38A8 23 18 DF FD E1 CD 00 AF : 74
38B0 DD 21 BD 38 FD E5 CD 04 : A6
38B8 B0 FD E1 FD E9 06 00 00 : 7A
38C0 00 00 7F 02 C7 00 0A 00 : 52
38C8 01 02 03 04 05 06 07 07 : 23
38D0 02 02 09 07 02 01 09 07 : 27
38D8 02 00 09 0F FD E1 D1 21 : EA
38E0 ED 38 73 23 D1 73 DD 21 : FD
38E8 EC 38 18 C8 07 02 00 0F : 1C
38F0 FD E1 D1 7B FE 03 30 10 : 6B
38F8 32 10 39 DD 21 0E 39 FD : BD
----------------------------------
SUM: 19 02 AB 67 AA A6 94 1C DFDB

3900 E5 CD 04 B0 FD E1 18 DE : 3A
3908 DD 21 CF 38 18 F1 07 02 : 17
3910 00 09 0F FD E1 21 FC 39 : 4C
3918 36 0F 06 08 2B D1 73 10 : D2
3920 FB 2B 36 0A FD E5 DD 21 : 46
```

THE SENTINEL

```
3928 F3 39 CD 04 B0 FD E1 FD : 88
3930 E9 58 CB 23 16 00 19 23 : 81
3938 36 0F D1 2B 72 2B 73 10 : 61
3940 F9 18 E1 FD E1 21 F3 39 : 1D
3948 36 06 06 04 18 E3 FD E1 : 1F
3950 21 F3 39 36 00 23 36 02 : DE
3958 06 04 18 D5 FD E1 21 F3 : E9
3960 39 36 01 06 06 18 CA FD : 5B
3968 E1 21 F3 39 36 02 06 04 : 70
3970 18 BF FD E1 21 85 39 D1 : 65
3978 72 2B 73 D1 2B 72 2B 73 : 1C
-----------------------------------
SUM: FF 27 23 46 D4 EA 53 CE C9EE

3980 FD E9 FF FF FF FF FD E1 : C0
3988 21 F3 39 36 04 06 04 23 : B4
3990 C5 11 82 39 01 04 00 EB : 81
3998 ED B0 EB C1 2B 18 92 FD : 1B
39A0 E1 21 F3 39 36 03 06 06 : 73
39A8 18 E5 FD E1 21 F3 39 36 : 5E
39B0 05 06 03 18 DA FD E1 21 : FF
39B8 F3 39 36 00 23 36 01 06 : C2
39C0 02 C3 31 39 FD E1 DD E1 : CB
39C8 C3 B4 38 FD E1 21 F3 39 : DA
39D0 36 08 C1 D1 23 73 23 72 : FB
39D8 23 71 23 70 23 36 0F FD : 8C
39E0 E5 DD 21 F3 39 CD 04 B0 : 90
39E8 FD E1 3A 02 C2 6F 26 00 : 71
39F0 E5 FD E9 00 00 00 00 00 : CB
39F8 00 00 00 00 00 00 00 00 : 00
-----------------------------------
SUM: A6 8D 5F CD A2 31 E0 88 42F5

3A00 00 00 00 00 00 00 00 00 : 00
3A08 00 00 00 00 00 00 00 00 : 00
3A10 00 00 00 3E 0C CD F4 1F : 2A
3A18 CD E2 1F 2A 2A 20 53 74 : 09
3A20 61 63 6B 20 43 6F 6D 70 : DE
3A28 69 6C 65 72 20 76 65 72 : 19
3A30 20 31 2E 30 20 2A 2A 0D : 30
3A38 00 CD E2 1F 54 45 58 54 : 13
3A40 20 41 44 44 52 45 53 53 : 26
3A48 20 20 20 3A 00 CD E5 3A : 86
3A50 D8 22 91 3E CD E2 1F 4F : E6
3A58 42 4A 45 43 54 20 41 44 : 0D
3A60 44 52 45 53 53 20 3A 00 : DB
3A68 CD E5 3A 38 A6 22 93 3E : BD
3A70 CD E2 1F 56 41 52 49 41 : 41
3A78 42 4C 45 20 54 4F 50 20 : 06
-----------------------------------
SUM: 31 E1 1C 49 0E 38 99 95 E585

3A80 20 20 3A 00 CD E5 3A 38 : 9E
3A88 8A 22 95 3E CD E2 1F 53 : A0
3A90 54 41 43 4B 20 54 4F 50 : 36
3A98 20 20 20 20 20 20 3A 00 : FA
3AA0 CD E5 3A DA 13 3A 22 97 : CC
3AA8 3E CD E2 1F 52 45 54 5F : 56
3AB0 53 54 41 43 4B 20 54 4F : 39
3AB8 50 20 20 3A 00 CD E5 3A : B6
3AC0 DA 13 3A 22 99 3E CD E2 : CF
3AC8 1F 4F 46 46 53 45 54 20 : 06
3AD0 41 44 44 52 45 53 53 20 : 26
3AD8 3A 00 CD E5 3A DA 13 3A : 4D
3AE0 22 9B 3E 18 15 ED 5B 76 : E6
3AE8 1F CD D3 1F 1A FE 1B 37 : 48
3AF0 C8 01 10 00 EB 09 EB C3 : 7B
3AF8 B2 1F CD E2 1F 50 41 53 : 83
-----------------------------------
SUM: FB F7 2E D7 2E 9B BA 79 0EA1

3B00 53 20 31 0D 00 CD 45 3B : FE
3B08 CD E2 1F 50 41 53 53 20 : 25
3B10 32 0D 00 CD 45 3B CD E2 : 3B
3B18 1F 4F 42 4A 45 43 54 20 : F6
3B20 45 4E 44 3A 0D 00 21 52 : 91
3B28 32 FD 36 00 C3 FD 75 01 : 9B
3B30 FD 74 02 FD 23 FD 23 FD : B0
3B38 23 FD E5 E1 CD BE 1F CD : 5D
3B40 EB 1F C3 FA 1F AF 32 9D : 64
3B48 3E DD 2A 91 3E FD 2A 93 : CE
3B50 3E ED 4B 9B 3E FD 09 21 : 76
3B58 0D 30 FD 36 00 CD FD 75 : AF
3B60 01 FD 74 02 FD 23 FD 23 : B4
3B68 FD 23 FD 36 00 ED FD 36 : 73
3B70 01 73 21 00 30 FD 75 02 : 39
3B78 FD 74 03 FD 36 04 31 2A : 06
-----------------------------------
SUM: 78 3A BD 1D 89 DD 93 C5 E2A4

3B80 97 3E FD 75 05 FD 74 06 : C3
3B88 FD 36 07 21 2A 99 3E FD : 59
3B90 75 08 FD 74 09 FD 36 0A : 34
3B98 22 21 02 30 FD 75 0B FD : EF
3BA0 74 0C 01 0D 00 FD 09 DD : 71
3BA8 7E 00 DD 23 B7 C8 FE FF : FA
3BB0 20 1E 3A 9D 3E B7 28 EF : 21
3BB8 FD E5 E1 ED 4B 9B 3E B7 : 8B
3BC0 ED 42 EB E1 73 23 72 3A : 3D
3BC8 9D 3E 3D 32 9D 3E 18 E2 : 1F
3BD0 FE 88 CA 43 3C FE 89 CA : 20
3BD8 E5 3C FE 8A CA 8A 3C FE : 37
3BE0 8B CA 65 3C FE 8C CA E7 : 31
3BE8 3D FE 8D CA 6A 3D FE 8E : C5
3BF0 CA 37 3D FE 8F CA 0A 3D : DC
3BF8 FE 90 CA A1 3D FE 91 CA : 8F
-----------------------------------
SUM: 37 7F E5 79 BF 99 12 EC 7013

3C00 C4 3D FE 92 CA 82 3D FE : 18
3C08 93 CA 50 3D FE 94 CA 10 : 56
3C10 3D FE 23 CA 1A 3E FE 10 : 8E
3C18 CA 30 3E FE 0E CA 45 3E : 91
3C20 FE 01 CA 4E 3E 6F 26 00 : EA
3C28 29 01 9E 3E 09 FD 36 00 : 42
3C30 CD 7E FD 77 01 23 7E FD : 5E
3C38 77 02 FD 23 FD 23 FD 23 : D9
3C40 C3 A7 3B FD E5 E1 ED 4B : A0
3C48 9B 3E B7 ED 42 EB DD 6E : F5
3C50 00 DD 66 01 29 DD 23 DD : 4A
3C58 23 7B CD 9A 1F 23 7A CD : 8E
3C60 9A 1F C3 A7 3B FD 36 00 : 91
3C68 C3 DD 6E 00 DD 66 01 29 : 7B
3C70 DD 23 DD 23 CD 94 1F FD : 7D
3C78 77 01 23 CD 94 1F FD 77 : 8F
-----------------------------------
SUM: FB 14 67 D9 1D B2 DB 7C CE40

3C80 02 FD 23 FD 23 FD 23 C3 : 25
3C88 A7 3B FD 36 00 21 FD 23 : 56
3C90 FD E5 E1 ED 4B 9B 3E B7 : 8B
3C98 ED 42 01 06 00 09 FD 75 : B1
3CA0 00 FD 74 01 FD 36 02 E5 : 8C
3CA8 FD 23 FD 23 FD 23 FD 36 : 93
3CB0 00 C3 FD 23 FD E5 FD 23 : E5
3CB8 FD 23 DD 7E 00 FE 22 28 : C3
3CC0 09 FD 77 00 DD 23 FD 23 : 9D
3CC8 18 F0 FD 36 00 0D DD 23 : 48
3CD0 FD 23 FD E5 E1 ED 4B 9B : B6
3CD8 3E B7 B7 ED 42 D1 EB 73 : 0A
3CE0 23 72 C3 A7 3B FD 36 00 : 6D
3CE8 E1 FD 36 01 7C FD 36 02 : C6
3CF0 B5 FD 36 03 CA 01 04 00 : BA
3CF8 FD 09 FD E5 FD 23 FD 23 : 28
-----------------------------------
SUM: 9F A1 A1 83 E3 0A F6 F1 E06D

3D00 3A 9D 3E 3C 32 9D 3E C3 : 21
3D08 A7 3B CD 19 3D C3 A7 3B : AA
3D10 CD 19 3D CD 19 3D C3 A7 : B0
3D18 3B FD 36 00 21 DD 7E 00 : EA
3D20 FD 77 01 DD 7E 01 FD 77 : 45
3D28 02 FD 36 03 E5 DD 23 DD : FA
3D30 23 01 04 00 FD 09 C9 CD : C4
3D38 65 3E FD 36 00 2A FD 75 : 72
3D40 01 FD 74 02 FD 36 03 E5 : 8F
3D48 01 04 00 FD 09 C3 A7 3B : B0
3D50 CD 65 3E FD 36 00 2A FD : CA
3D58 75 01 FD 74 02 FD 36 03 : 1F
3D60 E5 23 23 01 04 00 FD 09 : 36
3D68 18 D0 CD 65 3E CD 75 3E : D8
3D70 FD 36 00 22 FD 75 01 FD : C5
3D78 74 02 01 03 00 FD 09 C3 : 43
-----------------------------------
SUM: 22 33 56 33 86 C0 92 62 83D1

3D80 A7 3B CD 65 3E 23 23 CD : 65
3D88 75 3E FD 36 00 22 FD 75 : 7A
3D90 01 FD 74 02 FD 36 03 E1 : 8B
3D98 2B 2B 01 04 00 FD 09 18 : 79
3DA0 CF CD 65 3E FD 36 00 2A : 9C
3DA8 FD 75 01 FD 74 02 FD 36 : 19
3DB0 03 23 FD 36 04 22 FD 75 : F1
3DB8 05 FD 74 06 01 07 00 FD : 81
3DC0 09 C3 A7 3B CD 65 3E FD : 1B
3DC8 36 00 2A FD 75 01 FD 74 : 44
3DD0 02 FD 36 03 2B FD 36 04 : 9A
3DD8 22 FD 75 05 FD 74 06 01 : 11
3DE0 07 00 FD 09 C3 A7 3B DD : 8F
3DE8 6E 00 DD 66 01 29 CD 94 : 3C
3DF0 1F 5F 23 CD 94 1F 57 FD : 75
3DF8 36 00 11 FD 73 01 FD 72 : 27
-----------------------------------
SUM: 49 1F A0 91 E6 A0 F9 63 7FD3

3E00 02 FD 36 03 CD 21 C8 31 : 1F
3E08 FD 75 04 FD 74 05 DD 23 : EC
3E10 DD 23 01 06 00 FD 09 C3 : D0
3E18 A7 3B 21 D5 31 FD 36 00 : 3C
3E20 C3 FD 75 01 FD 74 02 FD : A6
3E28 23 FD 23 FD 23 C3 A7 3B : 08
3E30 FD 36 00 E1 FD 36 01 E5 : 2D
3E38 FD 36 02 E5 FD 23 FD 23 : 5A
3E40 FD 23 C3 A7 3B FD 36 00 : F8
3E48 E1 FD 23 C3 A7 3B CD 75 : E8
3E50 3E FD 36 00 D1 FD 36 01 : 76
3E58 19 FD 36 02 E5 01 03 00 : 37
3E60 FD 09 C3 A7 3B DD 6E 00 : F6
3E68 DD 66 01 ED 4B 95 3E 09 : 58
3E70 DD 23 DD 23 C9 FD 7E FF : 43
3E78 FE E5 20 0E FD 7E FC FE : 86
-----------------------------------
SUM: 4D C7 09 D0 70 D3 ED D3 7039

3E80 C3 28 07 FE CD 28 03 FD : E5
3E88 2B C9 FD 36 00 E1 FD 23 : 28
3E90 C9 00 00 00 00 00 00 00 : C9
3E98 00 00 00 00 00 00 00 00 : 00
3EA0 00 00 20 30 2A 30 34 30 : 0E
3EA8 3E 30 48 30 53 30 63 30 : FC
3EB0 73 30 79 30 89 30 96 30 : CB
3EB8 A3 30 00 00 E1 30 00 00 : E4
3EC0 F3 30 FE 30 09 31 14 31 : D0
3EC8 2C 31 3A 31 43 31 4B 31 : B8
3ED0 53 31 5C 31 71 31 00 00 : B3
3ED8 00 00 A6 31 AF 31 BE 31 : A6
3EE0 00 00 C8 31 D5 31 00 00 : FF
3EE8 E1 31 EE 31 BA 32 DA 32 : 29
3EF0 E3 32 EE 32 F7 32 00 33 : 91
3EF8 08 33 10 33 19 33 22 33 : 1F
-----------------------------------
SUM: 49 A9 D3 4E BF 55 46 DB 8854

3F00 29 33 32 33 3C 33 07 32 : 69
3F08 21 32 44 33 4D 33 55 33 : D2
3F10 5E 33 6A 33 78 33 83 33 : 8F
3F18 8D 33 98 33 BF 33 D5 33 : 85
3F20 EF 33 44 34 56 34 6C 34 : C4
3F28 89 34 B8 34 C6 34 D6 34 : AD
3F30 07 35 12 35 B9 35 24 36 : CB
3F38 33 36 4A 36 68 36 84 36 : 41
3F40 95 36 9B 36 B0 36 C3 36 : 7B
3F48 D5 36 DD 36 EA 36 05 37 : 7A
3F50 1A 37 3E 37 47 37 50 37 : CB
3F58 5E 37 68 37 6E 37 AB 38 : BC
3F60 DC 38 F0 38 13 39 43 39 : 04
3F68 4E 39 5C 39 67 39 72 39 : 67
3F70 9F 39 86 39 AA 39 CB 39 : 7E
3F78 B5 39 C4 39 84 37 87 37 : 64
-----------------------------------
SUM: 47 5A 84 5C F4 5B 68 5D 0C73

3F80 8A 37 96 37 52 32 61 32 : A5
3F88 70 32 A1 31 E9 30 77 32 : 36
3F90 8D 32 AC 32 02 36 1D 35 : 27
3F98 6D 35 E8 35 76 36 7D 36 : 1E
3FA0 10 37 7D 37 16 34 00 00 : 45
3FA8 00 00 00 00 00 00       : 00
-----------------------------------
SUM: 04 07 48 06 C9 02 72 CF 874F
```

リスト2 ソースリスト1

```
3000                 1          ORG   $3000
3000                 2          OFFSET $B000
3000                 3 ;
3000                 4 #PRINT   EQU   $1FF4
3000                 5 #MPRINT  EQU   $1FE2
3000                 6 #MSG     EQU   $1FE8
3000                 7 #NL      EQU   $1FEB
3000                 8 #GETL    EQU   $1FD3
3000                 9 #PRTHL   EQU   $1FBE
3000                10 #HLHEX   EQU   $1FB2
3000                11 #KBFAD   EQU   $1F76
3000                12 #PAUSE   EQU   $1FC7
3000                13 #BELL    EQU   $1FC4
3000                14 #HOT     EQU   $1FFA
3000                15 #FILE    EQU   $1FA3
3000                16 #WOPEN   EQU   $1FAF
3000                17 #WRD     EQU   $1FAC
3000                18 #ROPEN   EQU   $2009
3000                19 #RDD     EQU   $1FA6
3000                20 #FPRINT  EQU   $1F9D
3000                21 #DTADR   EQU   $1F70
3000                22 #SIZE    EQU   $1F72
3000                23 #EXADR   EQU   $1F6E
3000                24 #LPSW    EQU   $1F7C
3000                25 #MSX     EQU   $1FE5
3000                26 #PRTHX   EQU   $1FC1
3000                27 #BREAK   EQU   $1FCD
3000                28 #GETKY   EQU   $1FD0
3000                29 #INKEY   EQU   $1FCA
3000                30 #FLGET   EQU   $2021
3000                31 #2HEX    EQU   $1FB5
3000                32 #CSR     EQU   $2018
3000                33 #POKE    EQU   $1F9A
3000                34 #PEEK    EQU   $1F94
3000                35 #WIDCH   EQU   $2030
3000                36 #LOC     EQU   $201E
3000                37 #WIDTH   EQU   $1F5C
3000                38 #SCRN    EQU   $201B
3000                39 #DIR     EQU   $2006
3000                40 @RET_SP  EQU   $AE00
3000                41 @VAR_SP  EQU   $AA00
3000                42 ;
3000                43 ; STACK Compiler Run Time Routine
3000                44 ;
3000 00 00          45 MAIN_SP  DW        0
3002 00 AE          46 RET_SP   DW        $AE00
3004 00 00          47 STK_WR   DW        0
3006 00 00          48 RND0     DW        0
3008 00             49 #A       DB        0
3009 00 00          50 #DE      DW        0
300B 00 00          51 #HL      DW        0
300D                52 ;
300D                53 @START
300D 32 08 30       54          LD    (#A),A
3010 ED 53 09       55          LD    (#DE),DE
3013 30
3014 22 0B 30       56          LD    (#HL),HL
3017 C9             57          RET
3018                58 ;
3018                59 ; ｴﾝｻﾞﾝ (1)
3018                60 ;
3018                61 @TASU
3018 DD E1          62          POP   IX
301A D1             63          POP   DE
301B E1             64          POP   HL
301C 19             65          ADD   HL,DE
301D E5             66          PUSH  HL
301E DD E9          67          JP    (IX)
3020                68 @HIKU
3020 DD E1          69          POP   IX
3022 D1             70          POP   DE
3023 E1             71          POP   HL
3024 B7             72          OR    A
3025 ED 52          73          SBC   HL,DE
3027 E5             74          PUSH  HL
3028 DD E9          75          JP    (IX)
302A                76 @MLT
302A DD E1          77          POP   IX
302C D1             78          POP   DE
302D E1             79          POP   HL
```

▶僕は森田です。光磁気ディスクってMOとか略されるんですね。ペンネーム，MOリタにしちゃおうかな……。とかいって。
森田　宣幸(19)宮城県

```
302E CD B0 30    80         CALL MLT
3031 E5          81         PUSH HL
3032 DD E9       82         JP   (IX)
3034             83 @DIV
3034 DD E1       84         POP  IX
3036 D1          85         POP  DE
3037 E1          86         POP  HL
3038 CD C3 30    87         CALL DIV
303B E5          88         PUSH HL
303C DD E9       89         JP   (IX)
303E             90 @MOD
303E DD E1       91         POP  IX
3040 D1          92         POP  DE
3041 E1          93         POP  HL
3042 CD C3 30    94         CALL DIV
3045 D5          95         PUSH DE
3046 DD E9       96         JP   (IX)
3048             97 @DIVMOD
3048 DD E1       98         POP  IX
304A D1          99         POP  DE
304B E1         100         POP  HL
304C CD C3 30   101         CALL DIV
304F E5         102         PUSH HL
3050 D5         103         PUSH DE
3051 DD E9      104         JP   (IX)
3053            105 @==
3053 DD E1      106         POP  IX
3055 E1         107         POP  HL
3056 D1         108         POP  DE
3057 B7         109         OR   A
3058 ED 52      110         SBC  HL,DE
305A 21 00 00   111         LD   HL,0
305D 20 01      112         JR   NZ,@==1
305F 23         113         INC  HL
3060            114 @==1
3060 E5         115         PUSH HL
3061 DD E9      116         JP   (IX)
3063            117 @<
3063 DD E1      118         POP  IX
3065 D1         119         POP  DE
3066 E1         120         POP  HL
3067            121 @<'
3067 B7         122         OR   A
3068 ED 52      123         SBC  HL,DE
306A 21 01 00   124         LD   HL,1
306D 38 01      125         JR   C,@<1
306F 2B         126         DEC  HL
3070            127 @<1
3070 E5         128         PUSH HL
3071 DD E9      129         JP   (IX)
3073            130 @>
3073 DD E1      131         POP  IX
3075 E1         132         POP  HL
3076 D1         133         POP  DE
3077 18 EE      134         JR   @<'
3079            135 @!=
3079 DD E1      136         POP  IX
307B E1         137         POP  HL
307C D1         138         POP  DE
307D B7         139         OR   A
307E ED 52      140         SBC  HL,DE
3080 21 01 00   141         LD   HL,1
3083 20 01      142         JR   NZ,@!=1
3085 2B         143         DEC  HL
3086            144 @!=1
3086 E5         145         PUSH HL
3087 DD E9      146         JP   (IX)
3089            147 @AND
3089 DD E1      148         POP  IX
308B D1         149         POP  DE
308C E1         150         POP  HL
308D 7D         151         LD   A,L
308E A3         152         AND  E
308F 6F         153         LD   L,A
3090 7C         154         LD   A,H
3091 A2         155         AND  D
3092 67         156         LD   H,A
3093 E5         157         PUSH HL
3094 DD E9      158         JP   (IX)
3096            159 @OR
3096 DD E1      160         POP  IX
3098 D1         161         POP  DE
3099 E1         162         POP  HL
309A 7D         163         LD   A,L
309B B3         164         OR   E
309C 6F         165         LD   L,A
309D 7C         166         LD   A,H
309E B2         167         OR   D
309F 67         168         LD   H,A
30A0 E5         169         PUSH HL
30A1 DD E9      170         JP   (IX)
30A3            171 @XOR
30A3 DD E1      172         POP  IX
30A5 D1         173         POP  DE
30A6 E1         174         POP  HL
30A7 7D         175         LD   A,L
30A8 AB         176         XOR  E
30A9 6F         177         LD   L,A
30AA 7C         178         LD   A,H
30AB AA         179         XOR  D
30AC 67         180         LD   H,A
30AD E5         181         PUSH HL
30AE DD E9      182         JP   (IX)
30B0            183 ;
30B0            184 ; HL=HL*DE
30B0            185 ;
30B0            186 MLT
30B0 4D         187         LD   C,L
30B1 44         188         LD   B,H
30B2 3E 10      189         LD   A,16
30B4 21 00 00   190         LD   HL,0
30B7            191 MLT1
30B7 29         192         ADD  HL,HL
30B8 CB 23      193         SLA  E
30BA CB 12      194         RL   D
30BC 30 01      195         JR   NC,MLT2
30BE 09         196         ADD  HL,BC
30BF            197 MLT2
30BF 3D         198         DEC  A
30C0 20 F5      199         JR   NZ,MLT1
30C2 C9         200         RET
30C3            201 ;
30C3            202 ; HL=HL/DE
30C3            203 ;
30C3            204 ; DE=HL mod DE
30C3            205 ;
30C3            206 DIV
30C3 4B         207         LD   C,E
30C4 42         208         LD   B,D
30C5 5D         209         LD   E,L
30C6 54         210         LD   D,H
30C7 3E 10      211         LD   A,16
30C9 21 00 00   212         LD   HL,0
30CC            213 DIV1
30CC CB 23      214         SLA  E
30CE CB 12      215         RL   D
30D0 ED 6A      216         ADC  HL,HL
30D2 E5         217         PUSH HL
30D3 B7         218         OR   A
30D4 ED 42      219         SBC  HL,BC
30D6 E1         220         POP  HL
30D7 38 03      221         JR   C,DIV2
30D9 ED 42      222         SBC  HL,BC
30DB 13         223         INC  DE
30DC            224 DIV2
30DC 3D         225         DEC  A
30DD 20 ED      226         JR   NZ,DIV1
30DF EB         227         EX   DE,HL
30E0 C9         228         RET
30E1            229 ;
30E1            230 ; STACK ｿｳｻ
30E1            231 ;
30E1            232 @SWAP1
30E1 DD E1      233         POP  IX
30E3 D1         234         POP  DE
30E4 E1         235         POP  HL
30E5 D5         236         PUSH DE
30E6 E5         237         PUSH HL
30E7 DD E9      238         JP   (IX)
30E9            239 @ROT
30E9 DD E1      240         POP  IX
30EB E1         241         POP  HL
30EC D1         242         POP  DE
30ED C1         243         POP  BC
30EE D5         244         PUSH DE
30EF E5         245         PUSH HL
30F0 C5         246         PUSH BC
30F1 DD E9      247         JP   (IX)
30F3            248 ;
30F3            249 ; FUNCITON 1
30F3            250 ;
30F3            251 @KEY
30F3 DD E1      252         POP  IX
30F5 CD CA 1F   253         CALL #INKEY
30F8 6F         254         LD   L,A
30F9 26 00      255         LD   H,0
30FB E5         256         PUSH HL
30FC DD E9      257         JP   (IX)
30FE            258 @GETKEY
30FE DD E1      259         POP  IX
3100 CD D0 1F   260         CALL #GETKY
3103 6F         261         LD   L,A
3104 26 00      262         LD   H,0
3106 E5         263         PUSH HL
3107 DD E9      264         JP   (IX)
3109            265 @FLGET
3109 DD E1      266         POP  IX
310B CD 21 20   267         CALL #FLGET
310E 6F         268         LD   L,A
310F 26 00      269         LD   H,0
3111 E5         270         PUSH HL
3112 DD E9      271         JP   (IX)
3114            272 @RND
3114 DD E1      273         POP  IX
3116 2A 06 30   274         LD   HL,(RND0)
3119 54 5D      275         LD   DE,HL
311B 19         276         ADD  HL,DE
311C 19         277         ADD  HL,DE
311D 7D         278         LD   A,L
311E 84         279         ADD  A,H
311F 67         280         LD   H,A
3120 85         281         ADD  A,L
3121 6F         282         LD   L,A
3122 11 54 00   283         LD   DE,$54
3125 19         284         ADD  HL,DE
3126 22 06 30   285         LD   (RND0),HL
3129 E5         286         PUSH HL
312A DD E9      287         JP   (IX)
312C            288 @SCRN
312C DD E1      289         POP  IX
312E D1         290         POP  DE
312F E1         291         POP  HL
3130 63         292         LD   H,E
3131 CD 1B 20   293         CALL #SCRN
3134 6F         294         LD   L,A
3135 26 00      295         LD   H,0
3137 E5         296         PUSH HL
3138 DD E9      297         JP   (IX)
313A            298 ;
313A            299 ; PRINT 1
313A            300 ;
313A            301 @HEX2
313A DD E1      302         POP  IX
313C E1         303         POP  HL
313D 7D         304         LD   A,L
313E CD C1 1F   305         CALL #PRTHX
3141 DD E9      306         JP   (IX)
3143            307 @HEX4
3143 DD E1      308         POP  IX
3145 E1         309         POP  HL
3146 CD BE 1F   310         CALL #PRTHL
3149 DD E9      311         JP   (IX)
314B            312 @PRINT
314B DD E1      313         POP  IX
314D E1         314         POP  HL
314E CD 73 38   315         CALL @DEC2
3151 DD E9      316         JP   (IX)
3153            317 @CHR
3153 DD E1      318         POP  IX
3155 E1         319         POP  HL
3156 7D         320         LD   A,L
3157 CD F4 1F   321         CALL #PRINT
315A DD E9      322         JP   (IX)
315C            323 @PRTS
315C DD E1      324         POP  IX
315E E1         325         POP  HL
315F            326 @PRTS1
315F 7E         327         LD   A,(HL)
3160 FE 22      328         CP   '"'
3162 28 0B      329         JR   Z,@PRTS2
3164 FE 0D      330         CP   $0D
3166 CA 6F 31   331         JP   Z,@PRTS2
3169 CD F4 1F   332         CALL #PRINT
316C 23         333         INC  HL
316D 18 F0      334         JR   @PRTS1
316F            335 @PRTS2
316F DD E9      336         JP   (IX)
3171            337 @COTR
3171 DD E1      338         POP  IX
3173 E1         339         POP  HL
3174            340 @COTR1
3174 7E         341         LD   A,(HL)
3175 FE 22      342         CP   '"'
3177 28 F6      343         JR   Z,@PRTS2
3179 FE 0D      344         CP   $0D
317B CA 6F 31   345         JP   Z,@PRTS2
317E 0E 1C      346         LD   C,$1C
3180 FE 52      347         CP   'R'
3182 28 16      348         JR   Z,@COTR2
3184 0C         349         INC  C
3185 FE 4C      350         CP   'L'
3187 28 11      351         JR   Z,@COTR2
3189 0C         352         INC  C
318A FE 55      353         CP   'U'
318C 28 0C      354         JR   Z,@COTR2
318E 0C         355         INC  C
318F FE 44      356         CP   'D'
3191 28 07      357         JR   Z,@COTR2
3193 0E 0C      358         LD   C,$0C
3195 FE 43      359         CP   'C'
3197 28 01      360         JR   Z,@COTR2
3199 0C         361         INC  C
319A            362 @COTR2
319A 79         363         LD   A,C
319B CD F4 1F   364         CALL #PRINT
319E 23         365         INC  HL
319F 18 D3      366         JR   @COTR1
31A1            367 @CR
31A1 3E 0D      368         LD   A,$0D
31A3 C3 F4 1F   369         JP   #PRINT
31A6            370 ;
31A6            371 ; ｿﾉﾀ 1
31A6            372 ;
31A6            373 @WIDCH
31A6 DD E1      374         POP  IX
31A8 E1         375         POP  HL
31A9 7D         376         LD   A,L
31AA CD 30 20   377         CALL #WIDCH
31AD DD E9      378         JP   (IX)
31AF            379 @BELL
31AF DD E1      380         POP  IX
31B1 E1         381         POP  HL
31B2 45         382         LD   B,L
31B3 78         383         LD   A,B
31B4 B7         384         OR   A
31B5 28 05      385         JR   Z,@BELL2
31B7            386 @BELL1
31B7 CD C4 1F   387         CALL #BELL
31BA 10 FB      388         DJNZ @BELL1
31BC            389 @BELL2
31BC DD E9      390         JP   (IX)
31BE            391 @LOCATE
31BE DD E1      392         POP  IX
31C0 D1         393         POP  DE
31C1 E1         394         POP  HL
31C2 63         395         LD   H,E
31C3 CD 1E 20   396         CALL #LOC
31C6 DD E9      397         JP   (IX)
31C8            398 ;
31C8            399 ; ｾｲｷﾞｮｹｲ 1
31C8            400 ;
31C8            401 @GOSUB
31C8 C1         402         POP  BC
31C9 2A 02 30   403         LD   HL,(RET_SP)
31CC 2B         404         DEC  HL
31CD 70         405         LD   (HL),B
31CE 2B         406         DEC  HL
31CF 71         407         LD   (HL),C
31D0 22 02 30   408         LD   (RET_SP),HL
31D3 EB         409         EX   DE,HL
31D4 E9         410         JP   (HL)
31D5            411 @RETURN
31D5 2A 02 30   412         LD   HL,(RET_SP)
31D8 5E         413         LD   E,(HL)
31D9 23         414         INC  HL
31DA 56         415         LD   D,(HL)
31DB 23         416         INC  HL
31DC 22 02 30   417         LD   (RET_SP),HL
31DF EB         418         EX   DE,HL
31E0 E9         419         JP   (HL)
31E1            420 @REPEAT
31E1 D1         421         POP  DE
31E2 2A 02 30   422         LD   HL,(RET_SP)
31E5 2B         423         DEC  HL
31E6 72         424         LD   (HL),D
31E7 2B         425         DEC  HL
31E8 73         426         LD   (HL),E
31E9 22 02 30   427         LD   (RET_SP),HL
31EC EB         428         EX   DE,HL
31ED E9         429         JP   (HL)
31EE            430 @UNTIL
31EE DD E1      431         POP  IX
31F0 E1         432         POP  HL
31F1 7C         433         LD   A,H
31F2 B5         434         OR   L
31F3 28 0A      435         JR   Z,UNTIL1
31F5 2A 02 30   436         LD   HL,(RET_SP)
31F8 23         437         INC  HL
31F9 23         438         INC  HL
31FA 22 02 30   439         LD   (RET_SP),HL
31FD DD E9      440         JP   (IX)
31FF            441 UNTIL1
31FF 2A 02 30   442         LD   HL,(RET_SP)
3202 5E         443         LD   E,(HL)
3203 23         444         INC  HL
3204 56         445         LD   D,(HL)
3205 EB         446         EX   DE,HL
3206 E9         447         JP   (HL)
3207            448 @DO
3207 DD E1      449         POP  IX
3209 E1         450         POP  HL
320A D1         451         POP  DE
320B ED 73 04   452         LD   (STK_WR),SP
320E 30
320F ED 7B 02   453         LD   SP,(RET_SP)
3212 30
3213 DD E5      454         PUSH IX
3215 E5         455         PUSH HL
3216 D5         456         PUSH DE
3217 ED 73 02   457         LD   (RET_SP),SP
321A 30
321B ED 7B 04   458         LD   SP,(STK_WR)
321E 30
321F DD E9      459         JP   (IX)
3221            460 @LOOP!
3221 DD E1      461         POP  IX
3223 ED 73 04   462         LD   (STK_WR),SP
3226 30
3227 ED 7B 02   463         LD   SP,(RET_SP)
322A 30
322B D1         464         POP  DE
322C E1         465         POP  HL
322D FD E1      466         POP  IY
322F 13         467         INC  DE
3230 B7         468         OR   A
3231 ED 52      469         SBC  HL,DE
3233 38 13      470         JR   C,@LOOP!1
3235 19         471         ADD  HL,DE
3236 FD E5      472         PUSH IY
3238 E5         473         PUSH HL
3239 D5         474         PUSH DE
323A ED 73 02   475         LD   (RET_SP),SP
323D 30
323E ED 7B 04   476         LD   SP,(STK_WR)
3241 30
3242 FD E5      477         PUSH IY
3244 DD E1      478         POP  IX
3246 DD E9      479         JP   (IX)
3248            480 @LOOP!1
3248 ED 73 02   481         LD   (RET_SP),SP
324B 30
324C ED 7B 04   482         LD   SP,(STK_WR)
324F 30
3250 DD E9      483         JP   (IX)
3252            484 @END
3252 ED 7B 00   485         LD   SP,(MAIN_SP)
3255 30
3256 3A 08 30   486         LD   A,(#A)
3259 ED 5B 09   487         LD   DE,(#DE)
325C 30
325D 2A 0B 30   488         LD   HL,(#HL)
3260 C9         489         RET
3261            490 @I?
3261 DD E1      491         POP  IX
3263 11 00 00   492         LD   DE,0
3266            493 I?
3266 2A 02 30   494         LD   HL,(RET_SP)
3269 19         495         ADD  HL,DE
326A 5E         496         LD   E,(HL)
326B 23         497         INC  HL
326C 56         498         LD   D,(HL)
326D D5         499         PUSH DE
326E DD E9      500         JP   (IX)
3270            501 @J?
3270 DD E1      502         POP  IX
3272 11 06 00   503         LD   DE,6
3275 18 EF      504         JR   I?
3277            505 @TR
3277 DD E1      506         POP  IX
3279 E1         507         POP  HL
327A ED 73 04   508         LD   (STK_WR),SP
327D 30
327E ED 7B 02   509         LD   SP,(RET_SP)
3281 30
3282 E5         510         PUSH HL
3283 ED 73 02   511         LD   (RET_SP),SP
3286 30
3287 ED 7B 04   512         LD   SP,(STK_WR)
328A 30
328B DD E9      513         JP   (IX)
328D            514 @FR
328D DD E1      515         POP  IX
328F 2A 02 30   516         LD   HL,(RET_SP)
3292 01 00 AE   517         LD   BC,@RET_SP
3295 B7         518         OR   A
3296 ED 42      519         SBC  HL,BC
3298 ED 73 04   520         LD   (STK_WR),SP
329B 30
329C ED 7B 02   521         LD   SP,(RET_SP)
329F 30
32A0 E1         522         POP  HL
32A1 ED 73 02   523         LD   (RET_SP),SP
32A4 30
32A5 ED 7B 04   524         LD   SP,(STK_WR)
32A8 30
32A9 E5         525         PUSH HL
32AA DD E9      526         JP   (IX)
32AC            527 @LEA
32AC DD E1      528         POP  IX
32AE 2A 02 30   529         LD   HL,(RET_SP)
32B1 5E         530         LD   E,(HL)
32B2 23         531         INC  HL
32B3 56         532         LD   D,(HL)
32B4 23         533         INC  HL
32B5 73         534         LD   (HL),E
32B6 23         535         INC  HL
32B7 72         536         LD   (HL),D
32B8 DD E9      537         JP   (IX)
32BA            538 ;
32BA            539 ; ﾏｼﾝｺﾞ ｹｲ
32BA            540 ;
32BA            541 @CALL
32BA DD E1      542         POP  IX
32BC C1         543         POP  BC
32BD 2A 0B 30   544         LD   HL,(#HL)
32C0 ED 5B 09   545         LD   DE,(#DE)
32C3 30
32C4 3A 08 30   546         LD   A,(#A)
32C7 ED 43 CC   547         LD   (@CALL1+1),BC
32CA 32
32CB            548 @CALL1
32CB CD 00 00   549         DB   $CD,0,0
32CE 22 0B 30   550         LD   (#HL),HL
32D1 ED 53 09   551         LD   (#DE),DE
32D4 30
32D5 32 08 30   552         LD   (#A),A
32D8 DD E9      553         JP   (IX)
32DA            554 @PUTA
32DA DD E1      555         POP  IX
32DC E1         556         POP  HL
32DD 7D         557         LD   A,L
32DE 32 08 30   558         LD   (#A),A
32E1 DD E9      559         JP   (IX)
32E3            560 @GETA
```

THE SENTINEL

▶メモリは４Mバイト，ハードディスクは40Mバイト。あとほしいのは生活費と時間である。
渡辺　久孝(23)岡山県

```
32E3 DD E1        561           POP  IX
32E5 3A 08 30     562           LD   A,(#A)
32E8 6F           563           LD   L,A
32E9 26 00        564           LD   H,0
32EB E5           565           PUSH HL
32EC DD E9        566           JP   (IX)
32EE              567 @PUTD
32EE DD E1        568           POP  IX
32F0 D1           569           POP  DE
32F1 ED 53 09     570           LD   (#DE),DE
32F4 30
32F5 DD E9        571           JP   (IX)
32F7              572 @GETD
32F7 DD E1        573           POP  IX
32F9 ED 5B 09     574           LD   DE,(#DE)
32FC 30
32FD D5           575           PUSH DE
32FE DD E9        576           JP   (IX)
3300              577 @PUTH
3300 DD E1        578           POP  IX
3302 E1           579           POP  HL
3303 22 0B 30     580           LD   (#HL),HL
3306 DD E9        581           JP   (IX)
3308              582 @GETH
3308 DD E1        583           POP  IX
330A 2A 0B 30     584           LD   HL,(#HL)
330D E5           585           PUSH HL
330E DD E9        586           JP   (IX)
3310              587 ;
3310              588 ; ﾒﾓﾘ ｿｳｻ 1
3310              589 ;
3310              590 @PEEKB
3310 DD E1        591           POP  IX
3312 E1           592           POP  HL
3313 5E           593           LD   E,(HL)
3314 16 00        594           LD   D,0
3316 D5           595           PUSH DE
3317 DD E9        596           JP   (IX)
3319              597 @PEEKW
3319 DD E1        598           POP  IX
331B E1           599           POP  HL
331C 5E           600           LD   E,(HL)
331D 23           601           INC  HL
331E 56           602           LD   D,(HL)
331F D5           603           PUSH DE
3320 DD E9        604           JP   (IX)
3322              605 @POKEB
3322 DD E1        606           POP  IX
3324 E1           607           POP  HL
3325 D1           608           POP  DE
3326 73           609           LD   (HL),E
3327 DD E9        610           JP   (IX)
3329              611 @POKEW
3329 DD E1        612           POP  IX
332B E1           613           POP  HL
332C D1           614           POP  DE
332D 73           615           LD   (HL),E
332E 23           616           INC  HL
332F 72           617           LD   (HL),D
3330 DD E9        618           JP   (IX)
3332              619 ;
3332              620 ; I/O ｿｳｻ
3332              621 ;
3332              622 @IN
3332 DD E1        623           POP  IX
3334 C1           624           POP  BC
3335 ED 58        625           IN   E,(C)
3337 16 00        626           LD   D,0
3339 D5           627           PUSH DE
333A DD E9        628           JP   (IX)
333C              629 @OUT
333C DD E1        630           POP  IX
333E E1           631           POP  HL
333F C1           632           POP  BC
3340 ED 69        633           OUT  (C),L
3342 DD E9        634           JP   (IX)
3344              635 ;
3344              636 ;
3344              637 @HIGH
3344 DD E1        638           POP  IX
3346 E1           639           POP  HL
3347 6C           640           LD   L,H
3348 26 00        641           LD   H,0
334A E5           642           PUSH HL
334B DD E9        643           JP   (IX)
334D              644 @LOW
334D DD E1        645           POP  IX
334F E1           646           POP  HL
3350 26 00        647           LD   H,0
3352 E5           648           PUSH HL
3353 DD E9        649           JP   (IX)
3355              650 @EX
3355 DD E1        651           POP  IX
3357 E1           652           POP  HL
3358 7D           653           LD   A,L
3359 6C           654           LD   L,H
335A 67           655           LD   H,A
335B E5           656           PUSH HL
335C DD E9        657           JP   (IX)
335E              658 @NOT
335E DD E1        659           POP  IX
3360 E1           660           POP  HL
3361 7D           661           LD   A,L
3362 2F           662           CPL
3363 6F           663           LD   L,A
3364 7C           664           LD   A,H
3365 2F           665           CPL
3366 67           666           LD   H,A
3367 E5           667           PUSH HL
3368 DD E9        668           JP   (IX)
336A              669 @ROR
336A DD E1        670           POP  IX
336C C1           671           POP  BC
336D E1           672           POP  HL
336E 41           673           LD   B,C
336F              674 ROR1
336F CB 3C        675           SRL  H
3371 CB 1D        676           RR   L
3373 10 FA        677           DJNZ ROR1
3375 E5           678           PUSH HL
3376 DD E9        679           JP   (IX)
3378              680 @ROL
3378 DD E1        681           POP  IX
337A C1           682           POP  BC
337B E1           683           POP  HL
337C 41           684           LD   B,C
337D              685 ROL1
337D 29           686           ADD  HL,HL
337E 10 FD        687           DJNZ ROL1
3380 E5           688           PUSH HL
3381 DD E9        689           JP   (IX)
3383              690 @CURX
3383 DD E1        691           POP  IX
3385 CD 18 20     692           CALL #CSR
3388 26 00        693           LD   H,0
338A E5           694           PUSH HL
338B DD E9        695           JP   (IX)
338D              696 @CURY
338D DD E1        697           POP  IX
338F CD 18 20     698           CALL #CSR
3392 6C           699           LD   L,H
3393 26 00        700           LD   H,0
3395 E5           701           PUSH HL
3396 DD E9        702           JP   (IX)
3398              703 @NEGATE
3398 DD E1        704           POP  IX
339A E1           705           POP  HL
339B CD A1 33     706           CALL NEGATE
339E E5           707           PUSH HL
339F DD E9        708           JP   (IX)
33A1              709 NEGATE
33A1 7D           710           LD   A,L
33A2 2F           711           CPL
33A3 6F           712           LD   L,A
33A4 7C           713           LD   A,H
33A5 2F           714           CPL
33A6 67           715           LD   H,A
33A7 23           716           INC  HL
33A8 C9           717           RET
33A9              718 NEGATE2
33A9 7B           719           LD   A,E
33AA 2F           720           CPL
33AB 5F           721           LD   E,A
33AC 7A           722           LD   A,D
33AD 2F           723           CPL
33AE 57           724           LD   D,A
33AF 7D           725           LD   A,L
33B0 2F           726           CPL
33B1 6F           727           LD   L,A
33B2 7C           728           LD   A,H
33B3 2F           729           CPL
33B4 67           730           LD   H,A
33B5 01 01 00     731           LD   BC,1
33B8 EB           732           EX   DE,HL
33B9 09           733           ADD  HL,BC
33BA EB           734           EX   DE,HL
33BB 0B           735           DEC  BC
33BC ED 4A        736           ADC  HL,BC
33BE C9           737           RET
33BF              738 ;
33BF              739 ; STRING ｹｲ
33BF              740 ;
33BF              741 @STRCPY
33BF DD E1        742           POP  IX
33C1 E1           743           POP  HL
33C2 D1           744           POP  DE
33C3              745 STRCPY1
33C3 1A           746           LD   A,(DE)
33C4 77           747           LD   (HL),A
33C5 FE 0D        748           CP   $0D
33C7 28 0A        749           JR   Z,STRCPY3
33C9 FE 22        750           CP   '"'
33CB 28 04        751           JR   Z,STRCPY2
33CD 23           752           INC  HL
33CE 13           753           INC  DE
33CF 18 F2        754           JR   STRCPY1
33D1              755 STRCPY2
33D1 36 0D        756           LD   (HL),$0D
33D3              757 STRCPY3
33D3 DD E9        758           JP   (IX)
33D5              759 @LEFT$
33D5 DD E1        760           POP  IX
33D7 C1           761           POP  BC
33D8 E1           762           POP  HL
33D9 D1           763           POP  DE
33DA              764 LEFT1
33DA 1A           765           LD   A,(DE)
33DB 77           766           LD   (HL),A
33DC FE 0D        767           CP   $0D
33DE 28 0D        768           JR   Z,LEFT3
33E0 FE 22        769           CP   '"'
33E2 28 07        770           JR   Z,LEFT2
33E4 23           771           INC  HL
33E5 13           772           INC  DE
33E6 0B           773           DEC  BC
33E7 78           774           LD   A,B
33E8 B1           775           OR   C
33E9 20 EF        776           JR   NZ,LEFT1
33EB              777 LEFT2
33EB 36 0D        778           LD   (HL),$0D
33ED              779 LEFT3
33ED DD E9        780           JP   (IX)
33EF              781 @RIGHT$
33EF DD E1        782           POP  IX
33F1 C1           783           POP  BC
33F2 FD E1        784           POP  IY
33F4 E1           785           POP  HL
33F5 5D           786           LD   E,L
33F6 54           787           LD   D,H
33F7              788 RIGHT1
33F7 7E           789           LD   A,(HL)
33F8 FE 22        790           CP   '"'
33FA 28 07        791           JR   Z,RIGHT2
33FC FE 0D        792           CP   $0D
33FE 28 03        793           JR   Z,RIGHT2
3400 23           794           INC  HL
3401 18 F4        795           JR   RIGHT1
3403              796 RIGHT2
3403 2B           797           DEC  HL
3404 E5           798           PUSH HL
3405 B7           799           OR   A
3406 ED 52        800           SBC  HL,DE
3408 E1           801           POP  HL
3409 28 05        802           JR   Z,RIGHT3
340B 0B           803           DEC  BC
340C 78           804           LD   A,B
340D B1           805           OR   C
340E 20 F3        806           JR   NZ,RIGHT2
3410              807 RIGHT3
3410 FD E5        808           PUSH IY
3412 D1           809           POP  DE
3413 EB           810           EX   DE,HL
3414 18 AD        811           JR   STRCPY1
3416              812 @MID$
3416 DD E1        813           POP  IX
3418 D9           814           EXX
3419 C1           815           POP  BC
341A D9           816           EXX
341B C1           817           POP  BC
341C E1           818           POP  HL
341D D1           819           POP  DE
341E              820 MID1
341E 0B           821           DEC  BC
341F 78           822           LD   A,B
3420 B1           823           OR   C
3421 28 08        824           JR   Z,MID2
3423 1A           825           LD   A,(DE)
3424 FE 22        826           CP   '"'
3426 FE 0D        827           CP   $0D
3428 13           828           INC  DE
3429 18 F3        829           JR   MID1
342B              830 MID2
342B D9           831           EXX
342C C5           832           PUSH BC
342D D9           833           EXX
342E C1           834           POP  BC
342F              835 MID3
342F 1A           836           LD   A,(DE)
3430 77           837           LD   (HL),A
3431 FE 22        838           CP   '"'
3433 28 0B        839           JR   Z,MID4
3435 FE 0D        840           CP   $0D
3437 28 07        841           JR   Z,MID4
3439 23           842           INC  HL
343A 13           843           INC  DE
343B 0B           844           DEC  BC
343C 78           845           LD   A,B
343D B1           846           OR   C
343E 20 EF        847           JR   NZ,MID3
3440              848 MID4
3440 36 0D        849           LD   (HL),$0D
3442 DD E9        850           JP   (IX)
3444              851 @STRCAT
3444 DD E1        852           POP  IX
3446 D1           853           POP  DE
3447 E1           854           POP  HL
3448              855 STRCAT1
3448 7E           856           LD   A,(HL)
3449 FE 0D        857           CP   $0D
344B CA C3 33     858           JP   Z,STRCPY1
344E FE 22        859           CP   '"'
3450 CA C3 33     860           JP   Z,STRCPY1
3453 23           861           INC  HL
3454 18 F2        862           JR   STRCAT1
3456              863 @STRLEN
3456 DD E1        864           POP  IX
3458 E1           865           POP  HL
3459 01 00 00     866           LD   BC,0
345C              867 STRLEN1
345C 7E           868           LD   A,(HL)
345D FE 0D        869           CP   $0D
345F 28 08        870           JR   Z,STRLEN2
3461 FE 22        871           CP   '"'
3463 28 04        872           JR   Z,STRLEN2
3465 23           873           INC  HL
3466 03           874           INC  BC
3467 18 F3        875           JR   STRLEN1
3469              876 STRLEN2
3469 C5           877           PUSH BC
346A DD E9        878           JP   (IX)
346C              879 @INSTR
346C DD E1        880           POP  IX
346E C1           881           POP  BC
346F E1           882           POP  HL
3470 11 01 00     883           LD   DE,1
3473              884 INSTR1
3473 7E           885           LD   A,(HL)
3474 B9           886           CP   C
3475 28 0F        887           JR   Z,INSTR3
3477 FE 0D        888           CP   $0D
3479 28 08        889           JR   Z,INSTR2
347B FE 22        890           CP   '"'
347D 28 04        891           JR   Z,INSTR2
347F 13           892           INC  DE
3480 23           893           INC  HL
3481 18 F0        894           JR   INSTR1
3483              895 INSTR2
3483 11 00 00     896           LD   DE,0
3486              897 INSTR3
3486 D5           898           PUSH DE
3487 DD E9        899           JP   (IX)
3489              900 @STRCMP
3489 DD E1        901           POP  IX
348B E1           902           POP  HL
348C D1           903           POP  DE
348D              904 STRCMP1
348D 1A           905           LD   A,(DE)
348E FE 0D        906           CP   $0D
3490 28 0C        907           JR   Z,STRCMP2
3492 FE 22        908           CP   '"'
3494 28 08        909           JR   Z,STRCMP2
3496 46           910           LD   B,(HL)
3497 90           911           SUB  B
3498 20 13        912           JR   NZ,STRCMP3
349A 23           913           INC  HL
349B 13           914           INC  DE
349C 18 EF        915           JR   STRCMP1
349E              916 STRCMP2
349E 01 00 00     917           LD   BC,0
34A1 7E           918           LD   A,(HL)
34A2 FE 0D        919           CP   $0D
34A4 28 0F        920           JR   Z,STRCMP4
34A6 FE 22        921           CP   '"'
34A8 28 0B        922           JR   Z,STRCMP4
34AA 0B           923           DEC  BC
34AB 18 08        924           JR   STRCMP4
34AD              925 STRCMP3
34AD 01 01 00     926           LD   BC,1
34B0 30 03        927           JR   NC,STRCMP4
34B2 01 FF FF     928           LD   BC,-1
34B5              929 STRCMP4
34B5 C5           930           PUSH BC
34B6 DD E9        931           JP   (IX)
34B8              932 ;
34B8              933 ; 32 Bit ｴﾝｻﾞﾝ
34B8              934 ;
34B8              935 @LTASU
34B8 DD E1        936           POP  IX
34BA C1           937           POP  BC
34BB D1           938           POP  DE
34BC E1           939           POP  HL
34BD 09           940           ADD  HL,BC
34BE EB           941           EX   DE,HL
34BF C1           942           POP  BC
34C0 ED 4A        943           ADC  HL,BC
34C2 E5           944           PUSH HL
34C3 D5           945           PUSH DE
34C4 DD E9        946           JP   (IX)
34C6              947 @LHIKU
34C6 DD E1        948           POP  IX
34C8 D1           949           POP  DE
34C9 C1           950           POP  BC
34CA E1           951           POP  HL
34CB B7           952           OR   A
34CC ED 52        953           SBC  HL,DE
34CE EB           954           EX   DE,HL
34CF E1           955           POP  HL
34D0 ED 42        956           SBC  HL,BC
34D2 E5           957           PUSH HL
34D3 D5           958           PUSH DE
34D4 DD E9        959           JP   (IX)
34D6              960 @LMLT
34D6 DD E1        961           POP  IX
34D8 D1           962           POP  DE
34D9 E1           963           POP  HL
34DA D9           964           EXX
34DB FD 21 00     965           LD   IY,0
34DE 00
34DF 21 00 00     966           LD   HL,0
34E2 D1           967           POP  DE
34E3 C1           968           POP  BC
34E4 D9           969           EXX
34E5 06 20        970           LD   B,32
34E7              971 LMLT1
34E7 CB 3C        972           SRL  H
34E9 CB 1D        973           RR   L
34EB CB 1A        974           RR   D
34ED CB 1B        975           RR   E
34EF D9           976           EXX
34F0 30 04        977           JR   NC,LMLT2
34F2 FD 19        978           ADD  IY,DE
34F4 ED 4A        979           ADC  HL,BC
34F6              980 LMLT2
34F6 CB 23        981           SLA  E
34F8 CB 12        982           RL   D
34FA CB 11        983           RL   C
34FC CB 10        984           RL   B
34FE D9           985           EXX
34FF 10 E6        986           DJNZ LMLT1
3501 D9           987           EXX
3502 E5           988           PUSH HL
3503 FD E5        989           PUSH IY
3505 DD E9        990           JP   (IX)
3507              991 @LDIV
3507 DD E1        992           POP  IX
3509 CD 2C 35     993           CALL LDIV
350C D9           994           EXX
350D C5           995           PUSH BC
350E D9           996           EXX
350F C5           997           PUSH BC
3510 DD E9        998           JP   (IX)
3512              999 @LMOD
3512 DD E1       1000           POP  IX
3514 CD 2C 35    1001           CALL LDIV
3517 D9          1002           EXX
3518 E5          1003           PUSH HL
3519 D9          1004           EXX
351A E5          1005           PUSH HL
351B DD E9       1006           JP   (IX)
351D             1007 @LDIVMD
351D DD E1       1008           POP  IX
351F CD 2C 35    1009           CALL LDIV
3522 D9          1010           EXX
3523 E5          1011           PUSH HL
3524 D9          1012           EXX
3525 E5          1013           PUSH HL
3526 D9          1014           EXX
3527 C5          1015           PUSH BC
3528 D9          1016           EXX
3529 C5          1017           PUSH BC
352A DD E9       1018           JP   (IX)
352C             1019 ;
352C             1020 ; BC'BC=BC'BC/DE'DE
352C             1021 ; HL'HL=BC'BC MOD DE'DE
352C             1022 ;
352C             1023 LDIV
352C FD E1       1024           POP  IY
352E D1          1025           POP  DE
352F 21 00 00    1026           LD   HL,0
3532 D9          1027           EXX
3533 D1          1028           POP  DE
3534 21 00 00    1029           LD   HL,0
3537 D9          1030           EXX
3538 C1          1031           POP  BC
3539 D9          1032           EXX
353A C1          1033           POP  BC
353B D9          1034           EXX
353C             1035 ;
353C 3E 20       1036           LD   A,32
353E             1037 LDIV1
353E F5          1038           PUSH AF
353F CB 21       1039           SLA  C
3541 CB 10       1040           RL   B
3543 D9          1041           EXX
3544 CB 11       1042           RL   C
3546 CB 10       1043           RL   B
3548 D9          1044           EXX
3549 ED 6A       1045           ADC  HL,HL
354B D9          1046           EXX
354C ED 6A       1047           ADC  HL,HL
354E D9          1048           EXX
354F B7          1049           OR   A
3550 ED 52       1050           SBC  HL,DE
3552 D9          1051           EXX
3553 ED 52       1052           SBC  HL,DE
3555 D9          1053           EXX
3556 38 0E       1054           JR   C,LDIV2
3558             1055 ;
3558 03          1056           INC  BC
3559 78          1057           LD   A,B
355A B1          1058           OR   C
355B 20 03       1059           JR   NZ,LDIV3
355D D9          1060           EXX
355E 03          1061           INC  BC
355F D9          1062           EXX
```

▶鈴鹿F1レースを見ていたのですが，鈴木亜久里さんって本名だったんですね。私はまたハンドル名かと思ってました（通信じゃないって）。　　白沢　桂一(22)千葉県

```
3560                1063 LDIV3
3560 F1             1064         POP  AF
3561 3D             1065         DEC  A
3562 20 DA          1066         JR   NZ,LDIV1
3564 FD E9          1067         JP   (IY)
3566                1068 LDIV2
3566 19             1069         ADD  HL,DE
3567 D9             1070         EXX
3568 ED 5A          1071         ADC  HL,DE
356A D9             1072         EXX
356B 18 F3          1073         JR   LDIV3
356D                1074 @DDIVMOD
356D DD E1          1075         POP  IX
356F C1             1076         POP  BC
3570 D1             1077         POP  DE
3571 E1             1078         POP  HL
3572 CD 7C 35       1079         CALL QUOT
3575 D9             1080         EXX
3576 D5             1081         PUSH DE
3577 D9             1082         EXX
3578 E5             1083         PUSH HL
3579 D5             1084         PUSH DE
357A DD E9          1085         JP   (IX)
357C                1086 ;
357C                1087 ; HLDE=HLDE/BC
357C                1088 ;
357C                1089 ; DE'=HLDE MOD BC
357C                1090 ;
357C                1091 QUOT
357C F5             1092         PUSH AF
357D C5             1093         PUSH BC
357E D9             1094         EXX
357F C1             1095         POP  BC
3580 21 00 00       1096         LD   HL,0
3583 11 00 00       1097         LD   DE,0
3586 D9             1098         EXX
3587 3E 20          1099         LD   A,32
3589                1100 QUOT1
3589 EB             1101         EX   DE,HL
358A 29             1102         ADD  HL,HL
358B EB             1103         EX   DE,HL
358C ED 6A          1104         ADC  HL,HL
358E D9             1105         EXX
358F EB             1106         EX   DE,HL
3590 ED 6A          1107         ADC  HL,HL
3592 EB             1108         EX   DE,HL
3593 ED 6A          1109         ADC  HL,HL
3595 D5             1110         PUSH DE
3596 E5             1111         PUSH HL
3597 CD AA 35       1112         CALL QUOTSB
359A E1             1113         POP  HL
359B D1             1114         POP  DE
359C 38 03          1115         JR   C,QUOT2
359E CD AA 35       1116         CALL QUOTSB
35A1                1117 QUOT2
35A1 D9             1118         EXX
35A2 38 01          1119         JR   C,QUOT3
35A4 1C             1120         INC  E
35A5                1121 QUOT3
35A5 3D             1122         DEC  A
35A6 20 E1          1123         JR   NZ,QUOT1
35A8 F1             1124         POP  AF
35A9 C9             1125         RET
35AA                1126 QUOTSB
35AA EB             1127         EX   DE,HL
35AB B7             1128         OR   A
35AC ED 42          1129         SBC  HL,BC
35AE EB             1130         EX   DE,HL
35AF D0             1131         RET  NC
35B0 67             1132         LD   H,A
35B1 7D             1133         LD   A,L
35B2 D6 01          1134         SUB  1
35B4 6F             1135         LD   L,A
35B5 7C             1136         LD   A,H
35B6 26 00          1137         LD   H,0
35B8 C9             1138         RET
35B9                1139 @DMLT
35B9 DD E1          1140         POP  IX
35BB C1             1141         POP  BC
35BC D1             1142         POP  DE
35BD 21 00 00       1143         LD   HL,0
35C0 D9             1144         EXX
35C1 11 00 00       1145         LD   DE,0
35C4 21 00 00       1146         LD   HL,0
35C7 D9             1147         EXX
35C8 3E 10          1148         LD   A,16
35CA                1149 DMLT1
35CA CB 38          1150         SRL  B
35CC CB 19          1151         RR   C
35CE 30 05          1152         JR   NC,DMLT2
35D0 19             1153         ADD  HL,DE
35D1 D9             1154         EXX
35D2 ED 5A          1155         ADC  HL,DE
35D4 D9             1156         EXX
35D5                1157 DMLT2
35D5 CB 23          1158         SLA  E
35D7 CB 12          1159         RL   D
35D9 D9             1160         EXX
35DA CB 13          1161         RL   E
35DC CB 12          1162         RL   D
35DE D9             1163         EXX
35DF 3D             1164         DEC  A
35E0 20 E8          1165         JR   NZ,DMLT1
35E2 D9             1166         EXX
35E3 E5             1167         PUSH HL
35E4 D9             1168         EXX
35E5 E5             1169         PUSH HL
35E6 DD E9          1170         JP   (IX)
35E8                1171 @MLT!
35E8 DD E1          1172         POP  IX
35EA C1             1173         POP  BC
35EB D1             1174         POP  DE
35EC 21 00 00       1175         LD   HL,0
35EF 79             1176         LD   A,C
35F0                1177 MLT!1
35F0 B7             1178         OR   A
35F1 28 0C          1179         JR   Z,MLT!3
35F3 CB 3F          1180         SRL  A
35F5 30 01          1181         JR   NC,MLT!2
35F7 19             1182         ADD  HL,DE
35F8                1183 MLT!2
35F8 CB 23          1184         SLA  E
35FA CB 12          1185         RL   D
35FC C3 F0 35       1186         JP   MLT!1
35FF                1187 MLT!3
35FF E5             1188         PUSH HL
3600 DD E9          1189         JP   (IX)
3602                1190 @CMP2
3602 DD E1          1191         POP  IX
3604 C1             1192         POP  BC
3605 D1             1193         POP  DE
3606 E1             1194         POP  HL
3607 B7             1195         OR   A
3608 ED 42          1196         SBC  HL,BC
360A 4D             1197         LD   C,L
360B 44             1198         LD   B,H
360C E1             1199         POP  HL
360D ED 52          1200         SBC  HL,DE
360F 38 09          1201         JR   C,CMP2_1
3611 7C             1202         LD   A,H
3612 B5             1203         OR   L
3613 B0             1204         OR   B
3614 B1             1205         OR   C
3615 20 08          1206         JR   NZ,CMP2_2
3617                1207 CMP2_3
3617 E5             1208         PUSH HL
3618 DD E9          1209         JP   (IX)
361A                1210 CMP2_1
361A 21 FF FF       1211         LD   HL,-1
361D 18 F8          1212         JR   CMP2_3
361F                1213 CMP2_2
361F 21 01 00       1214         LD   HL,1
3622 18 F3          1215         JR   CMP2_3
3624                1216 @CTL
3624 DD E1          1217         POP  IX
3626 E1             1218         POP  HL
3627 11 00 00       1219         LD   DE,0
362A CB 7C          1220         BIT  7,H
362C 28 01          1221         JR   Z,CTL1
362E 1B             1222         DEC  DE
362F                1223 CTL1
362F D5             1224         PUSH DE
3630 E5             1225         PUSH HL
3631 DD E9          1226         JP   (IX)
3633                1227 @ASCII
3633 DD E1          1228         POP  IX
3635 E1             1229         POP  HL
3636 11 00 00       1230         LD   DE,0
3639                1231 ASCII1
```

```
3639 7E             1232         LD   A,(HL)
363A 23             1233         INC  HL
363B FE 0D          1234         CP   $0D
363D 28 08          1235         JR   Z,ASCII2
363F FE 22          1236         CP   '"'
3641 28 04          1237         JR   Z,ASCII2
3643 53             1238         LD   D,E
3644 5F             1239         LD   E,A
3645 18 F2          1240         JR   ASCII1
3647                1241 ASCII2
3647 D5             1242         PUSH DE
3648 DD E9          1243         JP   (IX)
364A                1244 @F<
364A DD E1          1245         POP  IX
364C D1             1246         POP  DE
364D E1             1247         POP  HL
364E 01 01 00       1248         LD   BC,1
3651 CB 7A          1249         BIT  7,D
3653 20 0D          1250         JR   NZ,F<1
3655 CB 7C          1251         BIT  7,H
3657 20 06          1252         JR   NZ,F<END
3659                1253 F<4
3659 B7             1254         OR   A
365A ED 52          1255         SBC  HL,DE
365C 38 01          1256         JR   C,F<END
365E                1257 F<3
365E 0B             1258         DEC  BC
365F                1259 F<END
365F C5             1260         PUSH BC
3660 DD E9          1261         JP   (IX)
3662                1262 F<1
3662 CB 7C          1263         BIT  7,H
3664 28 F8          1264         JR   Z,F<3
3666                1265 F<2
3666 18 F1          1266         JR   F<4
3668                1267 @=0
3668 DD E1          1268         POP  IX
366A E1             1269         POP  HL
366B 11 01 00       1270         LD   DE,1
366E 7C             1271         LD   A,H
366F B5             1272         OR   L
3670 28 01          1273         JR   Z,@=01
3672 1B             1274         DEC  DE
3673                1275 @=01
3673 D5             1276         PUSH DE
3674 DD E9          1277         JP   (IX)
3676                1278 @INC#
3676 DD E1          1279         POP  IX
3678 E1             1280         POP  HL
3679 23             1281         INC  HL
367A E5             1282         PUSH HL
367B DD E9          1283         JP   (IX)
367D                1284 @DEC#
367D DD E1          1285         POP  IX
367F E1             1286         POP  HL
3680 2B             1287         DEC  HL
3681 E5             1288         PUSH HL
3682 DD E9          1289         JP   (IX)
3684                1290 @PRINT1
3684 DD E1          1291         POP  IX
3686 D1             1292         POP  DE
3687 21 00 00       1293         LD   HL,0
368A                1294 @PRINT11
368A CD A0 37       1295         CALL CVHLDE
368D 11 C4 37       1296         LD   DE,@CVBUF
3690 CD E8 1F       1297         CALL #MSG
3693 DD E9          1298         JP   (IX)
3695                1299 @PRINT2
3695 DD E1          1300         POP  IX
3697 D1             1301         POP  DE
3698 E1             1302         POP  HL
3699 18 EF          1303         JR   @PRINT11
369B                1304 @PRF
369B DD E1          1305         POP  IX
369D E1             1306         POP  HL
369E 11 00 00       1307         LD   DE,0
36A1 CB 7C          1308         BIT  7,H
36A3 28 08          1309         JR   Z,PRINTF1
36A5 3E 2D          1310         LD   A,"-"
36A7 CD F4 1F       1311         CALL #PRINT
36AA CD A1 33       1312         CALL NEGATE
36AD                1313 PRINTF1
36AD EB             1314         EX   DE,HL
36AE 18 DA          1315         JR   @PRINT11
36B0                1316 @PRF2
36B0 DD E1          1317         POP  IX
36B2 D1             1318         POP  DE
36B3 E1             1319         POP  HL
36B4 CB 7C          1320         BIT  7,H
36B6 28 D2          1321         JR   Z,@PRINT11
36B8 3E 2D          1322         LD   A,"-"
36BA CD F4 1F       1323         CALL #PRINT
36BD CD A9 33       1324         CALL NEGATE2
36C0 C3 8A 36       1325         JP   @PRINT11
36C3                1326 @STRW
36C3 DD E1          1327         POP  IX
36C5 E1             1328         POP  HL
36C6 D1             1329         POP  DE
36C7 E5             1330         PUSH HL
36C8 21 00 00       1331         LD   HL,0
36CB                1332 STRW1
36CB CD A0 37       1333         CALL CVHLDE
36CE E1             1334         POP  HL
36CF 11 C4 37       1335         LD   DE,@CVBUF
36D2 C3 C3 33       1336         JP   STRCPY1
36D5                1337 @STRL
36D5 DD E1          1338         POP  IX
36D7 C1             1339         POP  BC
36D8 D1             1340         POP  DE
36D9 E1             1341         POP  HL
36DA C5             1342         PUSH BC
36DB 18 EE          1343         JR   STRW1
36DD                1344 @HEXL
36DD DD E1          1345         POP  IX
36DF D1             1346         POP  DE
36E0 E1             1347         POP  HL
36E1 CD BE 1F       1348         CALL #PRTHL
36E4 EB             1349         EX   DE,HL
36E5 CD BE 1F       1350         CALL #PRTHL
36E8 DD E9          1351         JP   (IX)
36EA                1352 @VAL1
36EA DD E1          1353         POP  IX
36EC FD E1          1354         POP  IY
36EE FD 7E 00       1355         LD   A,(IY)
36F1 FE 2D          1356         CP   "-"
36F3 28 06          1357         JR   Z,@VAL1_1
36F5 CD D0 37       1358         CALL DECI
36F8                1359 @VAL1_2
36F8 E5             1360         PUSH HL
36F9 DD E9          1361         JP   (IX)
36FB                1362 @VAL1_1
36FB FD 23          1363         INC  IY
36FD CD D0 37       1364         CALL DECI
3700 CD A1 33       1365         CALL NEGATE
3703 18 F3          1366         JR   @VAL1_2
3705                1367 @VAL2
3705 DD E1          1368         POP  IX
3707 FD E1          1369         POP  IY
3709 CD EE 37       1370         CALL HLDEDECI
370C E5             1371         PUSH HL
370D D5             1372         PUSH DE
370E DD E9          1373         JP   (IX)
3710                1374 @VAL$
3710 DD E1          1375         POP  IX
3712 FD E1          1376         POP  IY
3714 CD 87 38       1377         CALL #HEX
3717 E5             1378         PUSH HL
3718 DD E9          1379         JP   (IX)
371A                1380 @INP$
371A DD E1          1381         POP  IX
371C CD 18 20       1382         CALL #CSR
371F 26 00          1383         LD   H,0
3721 ED 5B 76       1384         LD   DE,(#KBFAD)
3724 1F
3725 CD D3 1F       1385         CALL #GETL
3728 1A             1386         LD   A,(DE)
3729 FE 1B          1387         CP   $1B
372B 28 0C          1388         JR   Z,INP1
372D 19             1389         ADD  HL,DE
372E EB             1390         EX   DE,HL
372F E1             1391         POP  HL
3730                1392 INP2
3730 1A             1393         LD   A,(DE)
3731 B7             1394         OR   A
3732 28 06          1395         JR   Z,INP3
3734 77             1396         LD   (HL),A
3735 13             1397         INC  DE
3736 23             1398         INC  HL
3737 18 F7          1399         JR   INP2
```

```
3739                1400 INP1
3739 E1             1401         POP  HL
373A                1402 INP3
373A 36 0D          1403         LD   (HL),$0D
373C DD E9          1404         JP   (IX)
373E                1405 @TRANS1
373E DD E1          1406         POP  IX
3740 C1             1407         POP  BC
3741 D1             1408         POP  DE
3742 E1             1409         POP  HL
3743 ED B0          1410         LDIR
3745 DD E9          1411         JP   (IX)
3747                1412 @TRANS2
3747 DD E1          1413         POP  IX
3749 C1             1414         POP  BC
374A D1             1415         POP  DE
374B E1             1416         POP  HL
374C ED B8          1417         LDDR
374E DD E9          1418         JP   (IX)
3750                1419 @FILL
3750 DD E1          1420         POP  IX
3752 D1             1421         POP  DE
3753 C1             1422         POP  BC
3754 E1             1423         POP  HL
3755 0B             1424         DEC  BC
3756 73             1425         LD   (HL),E
3757 54             1426         LD   D,H
3758 5D             1427         LD   E,L
3759 13             1428         INC  DE
375A ED B0          1429         LDIR
375C DD E9          1430         JP   (IX)
375E                1431 @COPYL
375E DD E1          1432         POP  IX
3760 D1             1433         POP  DE
3761 E1             1434         POP  HL
3762 E5             1435         PUSH HL
3763 D5             1436         PUSH DE
3764 E5             1437         PUSH HL
3765 D5             1438         PUSH DE
3766 DD E9          1439         JP   (IX)
3768                1440 @DROPL
3768 DD E1          1441         POP  IX
376A E1             1442         POP  HL
376B E1             1443         POP  HL
376C DD E9          1444         JP   (IX)
376E                1445 @SWAPD
376E DD E1          1446         POP  IX
3770 D1             1447         POP  DE
3771 E1             1448         POP  HL
3772 D9             1449         EXX
3773 D1             1450         POP  DE
3774 E1             1451         POP  HL
3775 D9             1452         EXX
3776 E5             1453         PUSH HL
3777 D5             1454         PUSH DE
3778 D9             1455         EXX
3779 E5             1456         PUSH HL
377A D5             1457         PUSH DE
377B DD E9          1458         JP   (IX)
377D                1459 @BREAK
377D CD CD 1F       1460         CALL #BREAK
3780 C0             1461         RET  NZ
3781 C3 52 32       1462         JP   @END
3784                1463 ;
3784                1464 #LPTON  EQU  $1FD9
3784                1465 #LPTOF  EQU  $1FD6
3784                1466 #SDVSW  EQU  $2027
3784                1467 ;
3784                1468 @PRON
3784 C3 D9 1F       1469         JP   #LPTON
3787                1470 @PROFF
3787 C3 D6 1F       1471         JP   #LPTOF
378A                1472 @PEEK#
378A DD E1          1473         POP  IX
378C E1             1474         POP  HL
378D CD 94 1F       1475         CALL #PEEK
3790 6F             1476         LD   L,A
3791 26 00          1477         LD   H,0
3793 E5             1478         PUSH HL
3794 DD E9          1479         JP   (IX)
3796                1480 @POKE#
3796 DD E1          1481         POP  IX
3798 C1             1482         POP  BC
3799 E1             1483         POP  HL
379A 79             1484         LD   A,C
379B CD 9A 1F       1485         CALL #POKE
379E DD E9          1486         JP   (IX)
37A0                1487 ;
37A0                1488 CVHLDE
37A0 01 00 00       1489         LD   BC,0
37A3                1490 CVHLDE1
37A3 C5             1491         PUSH BC
37A4 01 0A 00       1492         LD   BC,10
37A7 CD 7C 35       1493         CALL QUOT
37AA D9             1494         EXX
37AB 3E 30          1495         LD   A,'0'
37AD 83             1496         ADD  A,E
37AE D9             1497         EXX
37AF C1             1498         POP  BC
37B0 03             1499         INC  BC
37B1 F5             1500         PUSH AF
37B2 7C             1501         LD   A,H
37B3 B5             1502         OR   L
37B4 B2             1503         OR   D
37B5 B3             1504         OR   E
37B6 20 EB          1505         JR   NZ,CVHLDE1
37B8 41             1506         LD   B,C
37B9 21 C4 37       1507         LD   HL,@CVBUF
37BC                1508 CVHLDE2
37BC F1             1509         POP  AF
37BD 77             1510         LD   (HL),A
37BE 23             1511         INC  HL
37BF 10 FB          1512         DJNZ CVHLDE2
37C1 36 0D          1513         LD   (HL),$0D
37C3 C9             1514         RET
37C4                1515 @CVBUF
37C4 00 00 00       1516         DS 12
37C7 00 00 00
37CA 00 00 00
37CD 00 00 00
37D0                1517 ;
37D0                1518 ; CONVERT DECIMAL TO HL
37D0                1519 ;
37D0                1520 DECI
37D0 21 00 00       1521         LD   HL,0
37D3                1522 DECI1
37D3 FD 7E 00       1523         LD   A,(IY)
37D6 FE 30          1524         CP   '0'
37D8 D8             1525         RET  C
37D9 FE 3A          1526         CP   '9'+1
37DB D0             1527         RET  NC
37DC FD 23          1528         INC  IY
37DE 29             1529         ADD  HL,HL       ;HL=HL * 10
37DF 54             1530         LD   D,H
37E0 5D             1531         LD   E,L
37E1 29             1532         ADD  HL,HL
37E2 29             1533         ADD  HL,HL
37E3 19             1534         ADD  HL,DE
37E4 D6 30          1535         SUB  '0'
37E6 85             1536         ADD  A,L
37E7 30 01          1537         JR   NC,DECI2
37E9 24             1538         INC  H
37EA                1539 DECI2
37EA 6F             1540         LD   L,A
37EB C3 D3 37       1541         JP   DECI1
37EE                1542 HLDEDECI
37EE 21 00 00       1543         LD   HL,0
37F1 11 00 00       1544         LD   DE,0
37F4                1545 HLDE_1
37F4 FD 7E 00       1546         LD   A,(IY)
37F7 FE 30          1547         CP   '0'
37F9 D8             1548         RET  C
37FA FE 3A          1549         CP   '9'+1
37FC D0             1550         RET  NC
37FD FD 23          1551         INC  IY
37FF D6 30          1552         SUB  '0'
3801 EB             1553         EX   DE,HL   ;HLDE *2
3802 29             1554         ADD  HL,HL
3803 EB             1555         EX   DE,HL
3804 ED 6A          1556         ADC  HL,HL
3806                1557 ;
3806 44             1558         LD   B,H
3807 4D             1559         LD   C,L
3808 C5             1560         PUSH BC
3809 42             1561         LD   B,D
380A 4B             1562         LD   C,E
380B                1563 ;
380B EB             1564         EX   DE,HL
380C 29             1565         ADD  HL,HL
```


▶９月号といい，11月号といい，表紙のCGを書いている人はブタが好きなんでしょうか？ブタを飼っていたりして。

西　良太(20)埼玉県

```
380D EB          1566          EX   DE,HL
380E ED 6A       1567          ADC  HL,HL
3810 EB          1568          EX   DE,HL
3811 29          1569          ADD  HL,HL
3812 EB          1570          EX   DE,HL
3813 ED 6A       1571          ADC  HL,HL
3815             1572 ;
3815 EB          1573          EX   DE,HL
3816 09          1574          ADD  HL,BC
3817 EB          1575          EX   DE,HL
3818 C1          1576          POP  BC
3819 ED 4A       1577          ADC  HL,BC
381B             1578 ;
381B 06 00       1579          LD   B,0
381D 4F          1580          LD   C,A
381E EB          1581          EX   DE,HL
381F 09          1582          ADD  HL,BC
3820 EB          1583          EX   DE,HL
3821 0E 00       1584          LD   C,0
3823 ED 4A       1585          ADC  HL,BC
3825 18 CD       1586          JR   HLDE_1
3827             1587 ;
3827             1588 ; CONVERT HL TO DECIMAL
3827             1589 ;
3827             1590 CVHLD
3827 DD E5       1591          PUSH IX
3829 DD 21 64    1592          LD   IX,DTBL
382C 38
382D FD 21 6E    1593          LD   IY,CVTBL
3830 38
3831 3E 04       1594          LD   A,4
3833 06 05       1595          LD   B,5
3835             1596 CVHLD1
3835 0E 2F       1597          LD   C,'0'-1
3837 DD 5E 00    1598          LD   E,(IX)
383A DD 56 01    1599          LD   D,(IX+1)
383D             1600 CVHLD2
383D 0C          1601          INC  C
383E B7          1602          OR   A
383F ED 52       1603          SBC  HL,DE
3841 30 FA       1604          JR   NC,CVHLD2
3843 19          1605          ADD  HL,DE
3844 B7          1606          OR   A
3845 C4 56 38    1607          CALL NZ,CVHLD3
3848 FD 71 00    1608          LD   (IY),C
384B DD 23       1609          INC  IX
384D DD 23       1610          INC  IX
384F FD 23       1611          INC  IY
3851 10 E2       1612          DJNZ CVHLD1
3853 DD E1       1613          POP  IX
3855 C9          1614          RET
3856             1615 CVHLD3
3856 3D          1616          DEC  A
3857 08          1617          EX   AF,AF'
3858 3E 30       1618          LD   A,'0'
385A B9          1619          CP   C
385B 20 04       1620          JR   NZ,CVHLD4
385D 08          1621          EX   AF,AF'
385E 0E 20       1622          LD   C,' '
3860 C9          1623          RET
3861             1624 CVHLD4
3861 08          1625          EX   AF,AF'
3862 AF          1626          XOR  A
3863 C9          1627          RET
3864             1628 DTBL
3864 10 27 E8    1629          DW   10000,1000,100,10,1
3867 03 64 00
386A 0A 00 01
386D 00
386E             1630 CVTBL
386E 00 00 00    1631          DS   5
3871 00 00
3873             1632 @DEC2
3873 CD 27 38    1633          CALL CVHLD
3876 FD 21 6E    1634          LD   IY,CVTBL
3879 38
387A 06 05       1635          LD   B,5
387C             1636 @DECI1
387C FD 7E 00    1637          LD   A,(IY)
387F CD F4 1F    1638          CALL #PRINT
3882 FD 23       1639          INC  IY
3884 10 F6       1640          DJNZ @DECI1
3886 C9          1641          RET
3887             1642 #HEX
3887 21 00 00    1643          LD   HL,0
388A             1644 HEX1
388A DD 7E 00    1645          LD   A,(IX)
388D FE 30       1646          CP   "0"
388F D8          1647          RET  C
3890 FE 47       1648          CP   "F"+1
3892 D0          1649          RET  NC
3893             1650 ;
3893 FE 3A       1651          CP   "9"+1
3895 38 05       1652          JR   C,HEX2
3897 FE 41       1653          CP   "A"
3899 D8          1654          RET  C
389A D6 07       1655          SUB  7
389C             1656 HEX2
389C D6 30       1657          SUB  30H
389E 29          1658          ADD  HL,HL
389F 29          1659          ADD  HL,HL
38A0 29          1660          ADD  HL,HL
38A1 29          1661          ADD  HL,HL
38A2 85          1662          ADD  A,L
38A3 30 01       1663          JR   NC,HEXSKIP
38A5 24          1664          INC  H
38A6             1665 HEXSKIP
38A6 6F          1666          LD   L,A
38A7 DD 23       1667          INC   IX
38A9 18 DF       1668          JR   HEX1
38AB             1669 ;
38AB             1670 ; GRAPHIC
38AB             1671 ;
```

```
38AB             1672 MAGIC   EQU  $B004
38AB             1673 MAINIT  EQU  $AF00
38AB             1674 ;
38AB             1675 @INIT
38AB FD E1       1676          POP  IY
38AD CD 00 AF    1677          CALL MAINIT
38B0 DD 21 BD    1678          LD   IX,INITDATA
38B3 38
38B4             1679 INIT1
38B4 FD E5       1680          PUSH IY
38B6 CD 04 B0    1681          CALL MAGIC
38B9 FD E1       1682          POP  IY
38BB FD E9       1683          JP   (IY)
38BD             1684 INITDATA
38BD 06          1685          DB   6
38BE 00 00 00    1686          DW   0,0,639,199
38C1 00 7F 02
38C4 C7 00
38C6 0A 00 01    1687          DB   $0A,0,1,2,3,4,5,6,7
38C9 02 03 04
38CC 05 06 07
38CF             1688 CLSDATA
38CF 07 02 02    1689          DB   7,2,2,9,7,2,1,9,7,2,0,9
38D2 09 07 02
38D5 01 09 07
38D8 02 00 09
38DB 0F          1690          DB   $0F
38DC             1691 @COL
38DC FD E1       1692          POP  IY
38DE D1          1693          POP  DE
38DF 21 ED 38    1694          LD   HL,COLDATA+1
38E2 73          1695          LD   (HL),E
38E3 23          1696          INC  HL
38E4 D1          1697          POP  DE
38E5 73          1698          LD   (HL),E
38E6             1699 COL1
38E6 DD 21 EC    1700          LD   IX,COLDATA
38E9 38
38EA 18 C8       1701          JR   INIT1
38EC             1702 COLDATA
38EC 07          1703          DB   7
38ED 02          1704          DB   2
38EE 00          1705          DB   0
38EF 0F          1706          DB   $0F
38F0             1707 @CLS
38F0 FD E1       1708          POP  IY
38F2 D1          1709          POP  DE
38F3 7B          1710          LD   A,E
38F4 FE 03       1711          CP   3
38F6 30 10       1712          JR   NC,CLS1
38F8 32 10 39    1713          LD   (CLSDATA1+2),A
38FB DD 21 0E    1714          LD   IX,CLSDATA1
38FE 39
38FF             1715 CLS2
38FF FD E5       1716          PUSH IY
3901 CD 04 B0    1717          CALL MAGIC
3904 FD E1       1718          POP  IY
3906 18 DE       1719          JR   COL1
3908             1720 CLS1
3908 DD 21 CF    1721          LD   IX,CLSDATA
390B 38
390C 18 F1       1722          JR   CLS2
390E             1723 CLSDATA1
390E 07          1724          DB   7
390F 02          1725          DB   2
3910 00          1726          DB   0
3911 09          1727          DB   9
3912 0F          1728          DB   $0F
3913             1729 @PALET
3913 FD E1       1730          POP  IY
3915 21 FC 39    1731          LD   HL,MAGICBUF+9
3918 36 0F       1732          LD   (HL),$0F
391A 06 08       1733          LD   B,8
391C             1734 PALET1
391C 2B          1735          DEC  HL
391D D1          1736          POP  DE
391E 73          1737          LD   (HL),E
391F 10 FB       1738          DJNZ PALET1
3921 2B          1739          DEC  HL
3922 36 0A       1740          LD   (HL),$0A
3924             1741 PALET2
3924 FD E5       1742          PUSH IY
3926 DD 21 F3    1743          LD   IX,MAGICBUF
3929 39
392A CD 04 B0    1744          CALL MAGIC
392D FD E1       1745          POP  IY
392F FD E9       1746          JP   (IY)
3931             1747 WDATA
3931 58          1748          LD   E,B
3932 CB 23       1749          SLA  E
3934 16 00       1750          LD   D,0
3936 19          1751          ADD  HL,DE
3937 23          1752          INC  HL
3938 36 0F       1753          LD   (HL),$0F
393A             1754 WDATA1
393A D1          1755          POP  DE
393B 2B          1756          DEC  HL
393C 72          1757          LD   (HL),D
393D 2B          1758          DEC  HL
393E 73          1759          LD   (HL),E
393F 10 F9       1760          DJNZ WDATA1
3941 18 E1       1761          JR   PALET2
3943             1762 @WIND
3943 FD E1       1763          POP  IY
3945 21 F3 39    1764          LD   HL,MAGICBUF
3948 36 06       1765          LD   (HL),6
394A 06 04       1766          LD   B,4
394C 18 E3       1767          JR   WDATA
394E             1768 @LINE
394E FD E1       1769          POP  IY
3950 21 F3 39    1770          LD   HL,MAGICBUF
3953 36 00       1771          LD   (HL),0
3955 23          1772          INC  HL
```

```
3956 36 02       1773          LD   (HL),2
3958 06 04       1774          LD   B,4
395A 18 D5       1775          JR   WDATA
395C             1776 @SLINE
395C FD E1       1777          POP  IY
395E 21 F3 39    1778          LD   HL,MAGICBUF
3961 36 01       1779          LD   (HL),1
3963 06 06       1780          LD   B,6
3965 18 CA       1781          JR   WDATA
3967             1782 @BOX
3967 FD E1       1783          POP  IY
3969 21 F3 39    1784          LD   HL,MAGICBUF
396C 36 02       1785          LD   (HL),2
396E 06 04       1786          LD   B,4
3970 18 BF       1787          JR   WDATA
3972             1788 @TILE
3972 FD E1       1789          POP  IY
3974 21 85 39    1790          LD   HL,TILEBUF+3
3977 D1          1791          POP  DE
3978 72          1792          LD   (HL),D
3979 2B          1793          DEC  HL
397A 73          1794          LD   (HL),E
397B D1          1795          POP  DE
397C 2B          1796          DEC  HL
397D 72          1797          LD   (HL),D
397E 2B          1798          DEC  HL
397F 73          1799          LD   (HL),E
3980 FD E9       1800          JP   (IY)
3982             1801 TILEBUF
3982 FF FF       1802          DW  $FFFF
3984 FF FF       1803          DW  $FFFF
3986             1804 @BOXFUL
3986 FD E1       1805          POP  IY
3988 21 F3 39    1806          LD   HL,MAGICBUF
398B 36 04       1807          LD   (HL),4
398D 06 04       1808          LD   B,4
398F             1809 BOXF1
398F 23          1810          INC  HL
3990 C5          1811          PUSH BC
3991 11 82 39    1812          LD   DE,TILEBUF
3994 01 04 00    1813          LD   BC,4
3997 EB          1814          EX   DE,HL
3998 ED B0       1815          LDIR
399A EB          1816          EX   DE,HL
399B C1          1817          POP  BC
399C 2B          1818          DEC  HL
399D 18 92       1819          JR   WDATA
399F             1820 @TRIANGLE
399F FD E1       1821          POP  IY
39A1 21 F3 39    1822          LD   HL,MAGICBUF
39A4 36 03       1823          LD   (HL),3
39A6 06 06       1824          LD   B,6
39A8 18 E5       1825          JR   BOXF1
39AA             1826 @CIRCLE
39AA FD E1       1827          POP  IY
39AC 21 F3 39    1828          LD   HL,MAGICBUF
39AF 36 05       1829          LD   (HL),5
39B1 06 03       1830          LD   B,3
39B3 18 DA       1831          JR   BOXF1
39B5             1832 @DOT
39B5 FD E1       1833          POP  IY
39B7 21 F3 39    1834          LD   HL,MAGICBUF
39BA 36 00       1835          LD   (HL),0
39BC 23          1836          INC  HL
39BD 36 01       1837          LD   (HL),1
39BF 06 02       1838          LD   B,2
39C1 C3 31 39    1839          JP   WDATA
39C4             1840 @MAGIC
39C4 FD E1       1841          POP  IY
39C6 DD E1       1842          POP  IX
39C8 C3 B4 38    1843          JP   INIT1
39CB             1844 @POINT
39CB FD E1       1845          POP  IY
39CD 21 F3 39    1846          LD   HL,MAGICBUF
39D0 36 08       1847          LD   (HL),8
39D2 C1          1848          POP  BC
39D3 D1          1849          POP  DE
39D4 23          1850          INC  HL
39D5 73          1851          LD   (HL),E
39D6 23          1852          INC  HL
39D7 72          1853          LD   (HL),D
39D8 23          1854          INC  HL
39D9 71          1855          LD   (HL),C
39DA 23          1856          INC  HL
39DB 70          1857          LD   (HL),B
39DC 23          1858          INC  HL
39DD 36 0F       1859          LD   (HL),$0F
39DF FD E5       1860          PUSH IY
39E1 DD 21 F3    1861          LD   IX,MAGICBUF
39E4 39
39E5 CD 04 B0    1862          CALL MAGIC
39E8 FD E1       1863          POP  IY
39EA 3A 02 C2    1864          LD   A,($C202)
39ED 6F          1865          LD   L,A
39EE 26 00       1866          LD   H,0
39F0 E5          1867          PUSH HL
39F1 FD E9       1868          JP   (IY)
39F3             1869 MAGICBUF
39F3 00 00 00    1870          DS   32
39F6 00 00 00
39F9 00 00 00
39FC 00 00 00
39FF 00 00 00
3A02 00 00 00
3A05 00 00 00
3A08 00 00 00
3A0B 00 00 00
3A0E 00 00 00
3A11 00 00
3A13             1871 NEXT
```

リスト3 ソースリスト2

```
3A13               1          ORG   NEXT
3A13               2          OFFSET $B000
3A13               3 ;
3A13               4 #LABEL  EQU 136
3A13               5 #IF     EQU 137
3A13               6 #STR    EQU 138
3A13               7 #GOTO   EQU 139
3A13               8 #GOSUB  EQU 140
3A13               9 #LETW   EQU 141
3A13              10 #HENW   EQU 142
3A13              11 #WORDT  EQU 143
3A13              12 #INC    EQU 144
3A13              13 #DEC    EQU 145
3A13              14 #LETL   EQU 146
3A13              15 #HENL   EQU 147
3A13              16 #LONGT  EQU 148
3A13              17 #COPY   EQU 16
3A13              18 #DROP   EQU 14
3A13              19 #TASU   EQU  1
3A13              20 #RETURN EQU 35
3A13              21 #CR     EQU 255
3A13              22 ;
3A13              23 ;  Stack Compiler ver 1.0
3A13              24 ;
3A13              25 TITLE
3A13 3E 0C        26          LD   A,$0C
3A15 CD F4 1F     27          CALL #PRINT
3A18 CD E2 1F     28          CALL #MPRINT
3A1B 2A 2A 20     29          DM "** Stack Compiler ver 1.0 **"
3A1E 53 74 61
3A21 63 6B 20
3A24 43 6F 6D
3A27 70 69 6C
3A2A 65 72 20
3A2D 76 65 72
3A30 20 31 2E
3A33 30 20 2A
3A36 2A
3A37 0D 00        30          DB $0D:00
3A39 CD E2 1F     31          CALL #MPRINT
3A3C 54 45 58     32          DM   "TEXT ADDRESS   :"  DB 0
3A3F 54 20 41
3A42 44 44 52
3A45 45 53 53
3A48 20 20 20
3A4B 3A 00
3A4D CD E5 3A     33          CALL KEYIN
3A50 D8           34          RET  C ;;
3A51 22 91 3E     35          LD   (TEXT),HL
3A54 CD E2 1F     36          CALL #MPRINT
3A57 4F 42 4A     37          DM   "OBJECT ADDRESS :"  DB 0
3A5A 45 43 54
3A5D 20 41 44
3A60 44 52 45
```

```
3A63 53 53 20
3A66 3A 00
3A68 CD E5 3A     38          CALL KEYIN
3A6B 38 A6        39          JR   C,TITLE
3A6D 22 93 3E     40          LD   (OBJECT),HL
3A70 CD E2 1F     41          CALL #MPRINT
3A73 56 41 52     42          DM   "VARIABLE TOP   :"  DB 0
3A76 49 41 42
3A79 4C 45 20
3A7C 54 4F 50
3A7F 20 20 20
3A82 3A 00
3A84 CD E5 3A     43          CALL KEYIN
3A87 38 8A        44          JR   C,TITLE
3A89 22 95 3E     45          LD   (VAR),HL
3A8C CD E2 1F     46          CALL #MPRINT
3A8F 53 54 41     47          DM   "STACK TOP      :"  DB 0
3A92 43 4B 20
3A95 54 4F 50
3A98 20 20 20
3A9B 20 20 20
3A9E 3A 00
3AA0 CD E5 3A     48          CALL KEYIN
3AA3 DA 13 3A     49          JP   C,TITLE
3AA6 22 97 3E     50          LD   (ST_TOP),HL
3AA9 CD E2 1F     51          CALL #MPRINT
3AAC 52 45 54     52          DM   "RET_STACK TOP  :" DB 0
3AAF 5F 53 54
3AB2 41 43 4B
3AB5 20 54 4F
3AB8 50 20 20
3ABB 3A 00
3ABD CD E5 3A     53          CALL KEYIN
3AC0 DA 13 3A     54          JP   C,TITLE
3AC3 22 99 3E     55          LD   (RET_TOP),HL
3AC6 CD E2 1F     56          CALL #MPRINT
3AC9 4F 46 46     57          DM   "OFFSET ADDRESS :"  DB 0
3ACC 53 45 54
3ACF 20 41 44
3AD2 44 52 45
3AD5 53 53 20
3AD8 3A 00
3ADA CD E5 3A     58          CALL KEYIN
3ADD DA 13 3A     59          JP   C,TITLE
3AE0 22 9B 3E     60          LD   (OFFSET),HL
3AE3 18 15        61          JR   MAIN
3AE5              62 KEYIN
3AE5 ED 5B 76     63          LD   DE,(#KBFAD)
3AE8 1F
3AE9 CD D3 1F     64          CALL #GETL
3AEC 1A           65          LD   A,(DE)
3AED FE 1B        66          CP   1BH
3AEF 37           67          SCF
```

```
3AF0 C8           68          RET Z
3AF1 01 10 00     69          LD   BC,16
3AF4 EB 09 EB     70          EX   DE,HL  ADD  HL,BC  EX  DE,HL
3AF7 C3 B2 1F     71          JP   #HLHEX
3AFA              72 MAIN
3AFA CD E2 1F     73          CALL #MPRINT
3AFD 50 41 53     74          DM   "PASS 1"
3B00 53 20 31
3B03 0D 00        75          DB   $0D,0
3B05 CD 45 3B     76          CALL CCMAIN
3B08 CD E2 1F     77          CALL #MPRINT
3B0B 50 41 53     78          DM   "PASS 2"
3B0E 53 20 32
3B11 0D 00        79          DB   $0D,0
3B13 CD 45 3B     80          CALL CCMAIN
3B16 CD E2 1F     81          CALL #MPRINT
3B19 4F 42 4A     82          DM   "OBJECT END:"
3B1C 45 43 54
3B1F 20 45 4E
3B22 44 3A
3B24 0D 00        83          DB   $0D,0
3B26 21 52 32     84          LD   HL,@END
3B29 FD 36 00     85          LD   (IY),$C3
3B2C C3
3B2D FD 75 01     86          LD   (IY+1),L
3B30 FD 74 02     87          LD   (IY+2),H
3B33 FD 23        88          INC  IY
3B35 FD 23        89          INC  IY
3B37 FD 23        90          INC  IY
3B39 FD E5        91          PUSH IY
3B3B E1           92          POP  HL
3B3C CD BE 1F     93          CALL #PRTHL
3B3F CD EB 1F     94          CALL #NL
3B42 C3 FA 1F     95          JP   $1FFA
3B45              96 CCMAIN
3B45 AF           97          XOR  A
3B46 32 9D 3E     98          LD   (IFCOUNT),A
3B49 DD 2A 91     99          LD   IX,(TEXT)
3B4C 3E
3B4D FD 2A 93    100          LD   IY,(OBJECT)
3B50 3E
3B51 ED 4B 9B    101          LD   BC,(OFFSET)
3B54 3E
3B55 FD 09       102          ADD  IY,BC
3B57 21 0D 30    103          LD   HL,@START
3B5A FD 36 00    104          LD   (IY),$CD
3B5D CD
3B5E FD 75 01    105          LD   (IY+1),L
3B61 FD 74 02    106          LD   (IY+2),H      ;CALL @START
3B64 FD 23       107          INC  IY
3B66 FD 23       108          INC  IY
3B68 FD 23       109          INC  IY
3B6A FD 36 00    110          LD    (IY),$ED     ;LD (MAIN_SP),SP
3B6D ED
```

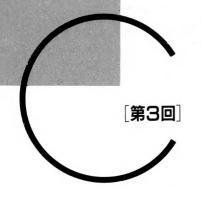
▶「理科系のGAME REVIEW」の上の荻窪圭さんのことば，とてもよかったです。なにかこう引き寄せられるものがありました。いっそのこと宗教団体作ってはどうですか。
梅津　長裕(17)東京都

```
3B6E FD 36 01    111            LD    (IY+1),$73
3B71 73
3B72 21 00 30    112            LD    HL,MAIN_SP
3B75 FD 75 02    113            LD    (IY+2),L
3B78 FD 74 03    114            LD    (IY+3),H
3B7B FD 36 04    115            LD    (IY+4),$31   ;LD SP,ST_TOP
3B7E 31
3B7F 2A 97 3E    116            LD    HL,(ST_TOP)
3B82 FD 75 05    117            LD    (IY+5),L
3B85 FD 74 06    118            LD    (IY+6),H
3B88 FD 36 07    119            LD    (IY+7),$21   ;LD HL,RET_TOP
3B8B 21
3B8C 2A 99 3E    120            LD    HL,(RET_TOP)
3B8F FD 75 08    121            LD    (IY+8),L
3B92 FD 74 09    122            LD    (IY+9),H
3B95 FD 36 0A    123            LD    (IY+10),$22 ;LD (RET_SP),HL
3B98 22
3B99 21 02 30    124            LD    HL,RET_SP
3B9C FD 75 0B    125            LD    (IY+11),L
3B9F FD 74 0C    126            LD    (IY+12),H
3BA2 01 0D 00    127            LD    BC,13
3BA5 FD 09       128            ADD   IY,BC
3BA7             129 CC1
3BA7 DD 7E 00    130            LD    A,(IX)
3BAA DD 23       131            INC   IX
3BAC B7          132            OR    A
3BAD C8          133            RET   Z            ;TEXT END
3BAE FE FF       134            CP    #CR          ;CR ?
3BB0 20 1E       135            JR    NZ,CC2
3BB2             136 CC3
3BB2 3A 9D 3E    137            LD    A,(IFCOUNT)
3BB5 B7          138            OR    A
3BB6 28 EF       139            JR    Z,CC1
3BB8 FD E5       140            PUSH  IY
3BBA E1          141            POP   HL
3BBB ED 4B 9B    142            LD    BC,(OFFSET)
3BBE 3E
3BBF B7          143            OR    A
3BC0 ED 42       144            SBC   HL,BC
3BC2 EB          145            EX    DE,HL
3BC3 E1          146            POP   HL
3BC4 73          147            LD    (HL),E
3BC5 23          148            INC   HL
3BC6 72          149            LD    (HL),D
3BC7 3A 9D 3E    150            LD    A,(IFCOUNT)
3BCA 3D          151            DEC   A
3BCB 32 9D 3E    152            LD    (IFCOUNT),A
3BCE 18 E2       153            JR    CC3
3BD0             154 CC2
3BD0 FE 88       155            CP    #LABEL
3BD2 CA 43 3C    156            JP    Z,!LABEL
3BD5 FE 89       157            CP    #IF
3BD7 CA E5 3C    158            JP    Z,!IF
3BDA FE 8A       159            CP    #STR
3BDC CA 8A 3C    160            JP    Z,!STR
3BDF FE 8B       161            CP    #GOTO
3BE1 CA 65 3C    162            JP    Z,!GOTO
3BE4 FE 8C       163            CP    #GOSUB
3BE6 CA E7 3D    164            JP    Z,!GOSUB
3BE9 FE 8D       165            CP    #LETW
3BEB CA 6A 3D    166            JP    Z,!LETW
3BEE FE 8E       167            CP    #HENW
3BF0 CA 37 3D    168            JP    Z,!HENW
3BF3 FE 8F       169            CP    #WORDT
3BF5 CA 0A 3D    170            JP    Z,!WORDT
3BF8 FE 90       171            CP    #INC
3BFA CA A1 3D    172            JP    Z,!INC
3BFD FE 91       173            CP    #DEC
3BFF CA C4 3D    174            JP    Z,!DEC
3C02 FE 92       175            CP    #LETL
3C04 CA 82 3D    176            JP    Z,!LETL
3C07 FE 93       177            CP    #HENL
3C09 CA 50 3D    178            JP    Z,!HENL
3C0C FE 94       179            CP    #LONGT
3C0E CA 10 3D    180            JP    Z,!LONGT
3C11 FE 23       181            CP    #RETURN
3C13 CA 1A 3E    182            JP    Z,!RETURN
3C16 FE 10       183            CP    #COPY
3C18 CA 30 3E    184            JP    Z,!COPY
3C1B FE 0E       185            CP    #DROP
3C1D CA 45 3E    186            JP    Z,!DROP
3C20 FE 01       187            CP    #TASU
3C22 CA 4E 3E    188            JP    Z,!TASU
3C25             189
3C25 6F          190            LD    L,A
3C26 26 00       191            LD    H,0
3C28 29          192            ADD   HL,HL
3C29 01 9E 3E    193            LD    BC,COMTBL
3C2C 09          194            ADD   HL,BC
3C2D FD 36 00    195            LD    (IY),$CD     ;CALL nn
3C30 CD
3C31 7E          196            LD    A,(HL)
3C32 FD 77 01    197            LD    (IY+1),A
3C35 23          198            INC   HL
3C36 7E          199            LD    A,(HL)
3C37 FD 77 02    200            LD    (IY+2),A
3C3A FD 23       201            INC   IY
3C3C FD 23       202            INC   IY
3C3E FD 23       203            INC   IY
3C40 C3 A7 3B    204            JP    CC1
3C43             205 !LABEL
3C43 FD E5       206            PUSH  IY
3C45 E1          207            POP   HL
3C46 ED 4B 9B    208            LD    BC,(OFFSET)
3C49 3E
3C4A B7          209            OR    A
3C4B ED 42       210            SBC   HL,BC
3C4D EB          211            EX    DE,HL
3C4E DD 6E 00    212            LD    L,(IX)
3C51 DD 66 01    213            LD    H,(IX+1)
3C54 29          214            ADD   HL,HL
3C55 DD 23       215            INC   IX
3C57 DD 23       216            INC   IX
3C59 7B          217            LD    A,E
3C5A CD 9A 1F    218            CALL  #POKE
3C5D 23          219            INC   HL
3C5E 7A          220            LD    A,D
3C5F CD 9A 1F    221            CALL  #POKE
3C62 C3 A7 3B    222            JP    CC1
3C65             223 !GOTO
3C65 FD 36 00    224            LD    (IY),$C3     ;JP  nn
3C68 C3
3C69 DD 6E 00    225            LD    L,(IX)
3C6C DD 66 01    226            LD    H,(IX+1)
3C6F 29          227            ADD   HL,HL
3C70 DD 23       228            INC   IX
3C72 DD 23       229            INC   IX
3C74 CD 94 1F    230            CALL  #PEEK
3C77 FD 77 01    231            LD    (IY+1),A
3C7A 23          232            INC   HL
3C7B CD 94 1F    233            CALL  #PEEK
3C7E FD 77 02    234            LD    (IY+2),A
3C81 FD 23       235            INC   IY
3C83 FD 23       236            INC   IY
3C85 FD 23       237            INC   IY
3C87 C3 A7 3B    238            JP    CC1
3C8A             239 !STR
3C8A FD 36 00    240            LD    (IY),$21   ;LD   HL,nn
3C8D 21
3C8E FD 23       241            INC   IY
3C90 FD E5       242            PUSH  IY
3C92 E1          243            POP   HL
3C93 ED 4B 9B    244            LD    BC,(OFFSET)
3C96 3E
3C97 B7          245            OR    A
3C98 ED 42       246            SBC   HL.BC
3C9A 01 06 00    247            LD    BC,6
3C9D 09          248            ADD   HL.BC
3C9E FD 75 00    249            LD    (IY),L
3CA1 FD 74 01    250            LD    (IY+1),H
3CA4 FD 36 02    251            LD    (IY+2),$E5 ;PUSH HL
3CA7 E5
3CA8 FD 23       252            INC   IY
3CAA FD 23       253            INC   IY
3CAC FD 23       254            INC   IY
3CAE FD 36 00    255            LD    (IY),$C3   ;JP  nn
3CB1 C3
3CB2 FD 23       256            INC   IY
3CB4 FD E5       257            PUSH  IY
3CB6 FD 23       258            INC   IY
3CB8 FD 23       259            INC   IY
3CBA             260 !STR1
3CBA DD 7E 00    261            LD    A,(IX)
3CBD FE 22       262            CP    '"'
3CBF 28 09       263            JR    Z,!STR2
3CC1 FD 77 00    264            LD   (IY),A
3CC4 DD 23       265            INC   IX
3CC6 FD 23       266            INC   IY
3CC8 18 F0       267            JR    !STR1
3CCA             268 !STR2
3CCA FD 36 00    269            LD    (IY),$0D
3CCD 0D
3CCE DD 23       270            INC   IX
3CD0 FD 23       271            INC   IY
3CD2             272
3CD2 FD E5       273            PUSH  IY
3CD4 E1          274            POP   HL
3CD5 ED 4B 9B    275            LD    BC,(OFFSET)
3CD8 3E
3CD9 B7          276            OR    A
3CDA B7 ED 42    277            SUB   HL,BC
3CDD D1          278            POP   DE
3CDE EB          279            EX    DE,HL
3CDF 73          280            LD    (HL),E
3CE0 23          281            INC   HL
3CE1 72          282            LD    (HL),D
3CE2 C3 A7 3B    283            JP    CC1
3CE5             284 !IF
3CE5 FD 36 00    285            LD    (IY),$E1    ;POP  HL
3CE8 E1
3CE9 FD 36 01    286            LD    (IY+1),$7C ;LD   A,H
3CEC 7C
3CED FD 36 02    287            LD    (IY+2),$B5 ;OR   L
3CF0 B5
3CF1 FD 36 03    288            LD    (IY+3),$CA ;JP   Z,nn
3CF4 CA
3CF5 01 04 00    289            LD    BC,4
3CF8 FD 09       290            ADD   IY,BC
3CFA FD E5       291            PUSH  IY
3CFC FD 23       292            INC   IY
3CFE FD 23       293            INC   IY
3D00 3A 9D 3E    294            LD    A,(IFCOUNT)
3D03 3C          295            INC   A
3D04 32 9D 3E    296            LD    (IFCOUNT),A
3D07 C3 A7 3B    297            JP    CC1
3D0A             298 !WORDT
3D0A CD 19 3D    299            CALL  !TEISUU
3D0D C3 A7 3B    300            JP    CC1
3D10             301 !LONGT
3D10 CD 19 3D    302            CALL  !TEISUU
3D13 CD 19 3D    303            CALL  !TEISUU
3D16 C3 A7 3B    304            JP    CC1
3D19             305 !TEISUU
3D19 FD 36 00    306            LD    (IY),$21    ;LD   HL,nn
3D1C 21
3D1D DD 7E 00    307            LD    A,(IX)
3D20 FD 77 01    308            LD    (IY+1),A
3D23 DD 7E 01    309            LD    A,(IX+1)
3D26 FD 77 02    310            LD    (IY+2),A
3D29 FD 36 03    311            LD    (IY+3),$E5 ;PUSH HL
3D2C E5
3D2D DD 23       312            INC   IX
3D2F DD 23       313            INC   IX
3D31 01 04 00    314            LD    BC,4
3D34 FD 09       315            ADD   IY,BC
3D36 C9          316            RET
3D37             317 !HENW
3D37 CD 65 3E    318            CALL  GETVAR
3D3A             319 !HENW1
3D3A FD 36 00    320            LD    (IY),$2A    ;LD   HL,(nn)
3D3D 2A
3D3E FD 75 01    321            LD    (IY+1),L
3D41 FD 74 02    322            LD    (IY+2),H
3D44 FD 36 03    323            LD    (IY+3),$E5 ;PUSH HL
3D47 E5
3D48 01 04 00    324            LD    BC,4
3D4B FD 09       325            ADD   IY,BC
3D4D C3 A7 3B    326            JP    CC1
3D50             327 !HENL
3D50 CD 65 3E    328            CALL  GETVAR
3D53 FD 36 00    329            LD    (IY),$2A    ;LD   HL,(nn)
3D56 2A
3D57 FD 75 01    330            LD    (IY+1),L
3D5A FD 74 02    331            LD    (IY+2),H
3D5D FD 36 03    332            LD    (IY+3),$E5 ;PUSH HL
3D60 E5
3D61 23          333            INC   HL
3D62 23          334            INC   HL
3D63 01 04 00    335            LD    BC,4
3D66 FD 09       336            ADD   IY,BC
3D68 18 D0       337            JR    !HENW1
3D6A             338 !LETW
3D6A CD 65 3E    339            CALL  GETVAR
3D6D CD 75 3E    340            CALL  OPT1
3D70             341 !LETW1
3D70 FD 36 00    342            LD    (IY),$22    ;LD   (nn),HL
3D73 22
3D74 FD 75 01    343            LD    (IY+1),L
3D77 FD 74 02    344            LD    (IY+2),H
3D7A 01 03 00    345            LD    BC,3
3D7D FD 09       346            ADD   IY,BC
3D7F C3 A7 3B    347            JP    CC1
3D82             348 !LETL
3D82 CD 65 3E    349            CALL  GETVAR
3D85 23          350            INC   HL
3D86 23          351            INC   HL
3D87 CD 75 3E    352            CALL  OPT1
3D8A FD 36 00    353            LD    (IY),$22    ;LD   (nn),HL
3D8D 22
3D8E FD 75 01    354            LD    (IY+1),L
3D91 FD 74 02    355            LD    (IY+2),H
3D94 FD 36 03    356            LD    (IY+3),$E1 ;POP  HL
3D97 E1
3D98 2B          357            DEC   HL
3D99 2B          358            DEC   HL
3D9A 01 04 00    359            LD    BC,4
3D9D FD 09       360            ADD   IY,BC
3D9F 18 CF       361            JR    !LETW1
3DA1             362 !INC
3DA1 CD 65 3E    363            CALL  GETVAR
3DA4 FD 36 00    364            LD    (IY),$2A    ;LD   HL,(nn)
3DA7 2A
3DA8 FD 75 01    365            LD    (IY+1),L
3DAB FD 74 02    366            LD    (IY+2),H
3DAE FD 36 03    367            LD    (IY+3),$23 ;INC  HL
3DB1 23
3DB2 FD 36 04    368            LD    (IY+4),$22 ;LD   (nn),HL
3DB5 22
3DB6 FD 75 05    369            LD    (IY+5),L
3DB9 FD 74 06    370            LD    (IY+6),H
3DBC 01 07 00    371            LD    BC,7
3DBF FD 09       372            ADD   IY,BC
3DC1 C3 A7 3B    373            JP    CC1
3DC4             374 !DEC
3DC4 CD 65 3E    375            CALL  GETVAR
3DC7 FD 36 00    376            LD    (IY),$2A    ;LD   HL,(nn)
3DCA 2A
3DCB FD 75 01    377            LD    (IY+1),L
3DCE FD 74 02    378            LD    (IY+2),H
3DD1 FD 36 03    379            LD    (IY+3),$2B ;DEC  HL
3DD4 2B
3DD5 FD 36 04    380            LD    (IY+4),$22 ;LD   (nn),HL
3DD8 22
3DD9 FD 75 05    381            LD    (IY+5),L
3DDC FD 74 06    382            LD    (IY+6),H
3DDF 01 07 00    383            LD    BC,7
3DE2 FD 09       384            ADD   IY,BC
3DE4 C3 A7 3B    385            JP    CC1
3DE7             386 !GOSUB
3DE7 DD 6E 00    387            LD    L,(IX)
3DEA DD 66 01    388            LD    H,(IX+1)
3DED 29          389            ADD   HL,HL
3DEE CD 94 1F    390            CALL  #PEEK
3DF1 5F          391            LD    E,A
3DF2 23          392            INC   HL
3DF3 CD 94 1F    393            CALL  #PEEK
3DF6 57          394            LD    D,A
3DF7             395
3DF7 FD 36 00    396            LD    (IY),$11    ;LD   DE,nn
3DFA 11
3DFB FD 73 01    397            LD    (IY+1),E
3DFE FD 72 02    398            LD    (IY+2),D
3E01 FD 36 03    399            LD    (IY+3),$CD ;CALL nn
3E04 CD
3E05 21 C8 31    400            LD    HL,@GOSUB
3E08 FD 75 04    401            LD    (IY+4),L
3E0B FD 74 05    402            LD    (IY+5),H
3E0E DD 23       403            INC   IX
3E10 DD 23       404            INC   IX
3E12 01 06 00    405            LD    BC,6
3E15 FD 09       406            ADD   IY,BC
3E17 C3 A7 3B    407            JP    CC1
3E1A             408 !RETURN
3E1A 21 D5 31    409            LD    HL,@RETURN
3E1D FD 36 00    410            LD    (IY),$C3     ;JP   @RETURN
3E20 C3
3E21 FD 75 01    411            LD    (IY+1),L
3E24 FD 74 02    412            LD    (IY+2),H
3E27 FD 23       413            INC   IY
3E29 FD 23       414            INC   IY
3E2B FD 23       415            INC   IY
3E2D C3 A7 3B    416            JP    CC1
3E30             417 !COPY
3E30 FD 36 00    418            LD    (IY),$E1     ;POP  HL
3E33 E1
3E34 FD 36 01    419            LD    (IY+1),$E5  ;PUSH HL
3E37 E5
3E38 FD 36 02    420            LD    (IY+2),$E5  ;PUSH HL
3E3B E5
3E3C FD 23       421            INC   IY
3E3E FD 23       422            INC   IY
3E40 FD 23       423            INC   IY
3E42 C3 A7 3B    424            JP    CC1
3E45             425 !DROP
3E45 FD 36 00    426            LD    (IY),$E1     ;POP  HL
3E48 E1
3E49 FD 23       427            INC   IY
3E4B C3 A7 3B    428            JP    CC1
3E4E             429 !TASU
3E4E CD 75 3E    430            CALL  OPT1
3E51 FD 36 00    431            LD    (IY),$D1     ;POP  DE
3E54 D1
3E55 FD 36 01    432            LD    (IY+1),$19 ;ADD  HL,DE
3E58 19
3E59 FD 36 02    433            LD    (IY+2),$E5 ;PUSH HL
3E5C E5
3E5D 01 03 00    434            LD    BC,3
3E60 FD 09       435            ADD   IY,BC
3E62 C3 A7 3B    436            JP    CC1
3E65             437 GETVAR
3E65 DD 6E 00    438            LD    L,(IX)
3E68 DD 66 01    439            LD    H,(IX+1)
3E6B ED 4B 95    440            LD    BC,(VAR)
3E6E 3E
3E6F 09          441            ADD   HL,BC
3E70 DD 23       442            INC   IX
3E72 DD 23       443            INC   IX
3E74 C9          444            RET
3E75             445 OPT1
3E75 FD 7E FF    446            LD    A,(IY-1)
3E78 FE E5       447            CP    $E5          ;PUSH HL?
3E7A 20 0E       448            JR    NZ,OPT1_1
3E7C FD 7E FC    449            LD    A,(IY-4)
3E7F FE C3       450            CP    $C3          ;JP ?
3E81 28 07       451            JR    Z,OPT1_1
3E83 FE CD       452            CP    $CD          ;CALL ?
3E85 28 03       453            JR    Z,OPT1_1
3E87 FD 2B       454            DEC   IY
3E89 C9          455            RET
3E8A             456 OPT1_1
3E8A FD 36 00    457            LD    (IY),$E1     ;POP  HL
3E8D E1
3E8E FD 23       458            INC   IY
3E90 C9          459            RET
3E91             460
3E91             461
3E91 00 00       462 TEXT    DW      0
3E93 00 00       463 OBJECT  DW      0
3E95 00 00       464 VAR     DW      0
3E97 00 00       465 ST_TOP  DW      0
3E99 00 00       466 RET_TOP DW      0
3E9B 00 00       467 OFFSET  DW      0
3E9D 00          468 IFCOUNT DB      0
3E9E             469 ;
3E9E             470 COMTBL
3E9E 00 00       471         DW    0
3EA0 00 00       472         DW    0          ; @TASU
3EA2 20 30       473         DW    @HIKU
3EA4 2A 30       474         DW    @MLT
3EA6 34 30       475         DW    @DIV
3EA8 3E 30       476         DW    @MOD
3EAA 48 30       477         DW    @DIVMOD
3EAC 53 30       478         DW    @==
3EAE 63 30       479         DW    @<
3EB0 73 30       480         DW    @>
3EB2 79 30       481         DW    @!=
3EB4 89 30       482         DW    @AND
3EB6 96 30       483         DW    @OR
3EB8 A3 30       484         DW    @XOR
3EBA             485 ;
3EBA 00 00       486         DW    0         ;@DROP
3EBC E1 30       487         DW    @SWAP1
3EBE 00 00       488         DW    0         ;@COPY
3EC0             489 ;
3EC0 F3 30       490         DW    @KEY
3EC2 FE 30       491         DW    @GETKEY
3EC4 09 31       492         DW    @FLGET
3EC6 14 31       493         DW    @RND
3EC8 2C 31       494         DW    @SCRN
3ECA             495 ;
3ECA 3A 31       496         DW    @HEX2
3ECC 43 31       497         DW    @HEX4
3ECE 4B 31       498         DW    @PRINT
3ED0 53 31       499         DW    @CHR
3ED2 5C 31       500         DW    @PRTS
3ED4 71 31       501         DW    @COTR
3ED6             502 ;
3ED6 00 00       503         DW    0          ;@INC
3ED8 00 00       504         DW    0         ;@DEC
3EDA A6 31       505         DW    @WIDCH
3EDC AF 31       506         DW    @BELL
3EDE BE 31       507         DW    @LOCATE
3EE0             508 ;
3EE0 00 00       509         DW    0         ;@GOTO
3EE2 C8 31       510         DW    @GOSUB
3EE4 D5 31       511         DW    @RETURN
3EE6 00 00       512         DW    0         ;@IF
3EE8 E1 31       513         DW    @REPEAT
3EEA EE 31       514         DW    @UNTIL
3EEC             515 ;
3EEC BA 32       516         DW    @CALL
3EEE DA 32       517         DW    @PUTA
3EF0 E3 32       518         DW    @GETA
3EF2 EE 32       519         DW    @PUTD
3EF4 F7 32       520         DW    @GETD
3EF6 00 33       521         DW    @PUTH
3EF8 08 33       522         DW    @GETH
3EFA             523 ;
3EFA 10 33       524         DW    @PEEKB
3EFC 19 33       525         DW    @PEEKW
3EFE 22 33       526         DW    @POKEB
3F00 29 33       527         DW    @POKEW
3F02             528 ;
3F02 32 33       529         DW    @IN
3F04 3C 33       530         DW    @OUT
3F06             531 ;
3F06 07 32       532         DW    @DO
3F08 21 32       533         DW    @LOOP!
3F0A             534 ;
3F0A 44 33       535         DW    @HIGH
3F0C 4D 33       536         DW    @LOW
3F0E 55 33       537         DW    @EX
3F10 5E 33       538         DW    @NOT
3F12 6A 33       539         DW    @ROR
3F14 78 33       540         DW    @ROL
3F16 83 33       541         DW    @CURX
3F18 8D 33       542         DW    @CURY
3F1A 98 33       543         DW    @NEGATE
3F1C             544 ;
3F1C BF 33       545         DW    @STRCPY
3F1E D5 33       546         DW    @LEFT$
3F20 EF 33       547         DW    @RIGHT$
3F22 44 34       548         DW    @STRCAT
3F24 56 34       549         DW    @STRLEN
3F26 6C 34       550         DW    @INSTR
3F28 89 34       551         DW    @STRCMP
3F2A             552 ;
3F2A B8 34       553         DW    @LTASU
3F2C C6 34       554         DW    @LHIKU
3F2E D6 34       555         DW    @LMLT
3F30 07 35       556         DW    @LDIV
3F32 12 35       557         DW    @LMOD
3F34 B9 35       558         DW    @DMLT
3F36 24 36       559         DW    @CTL
3F38             560 ;
3F38 33 36       561         DW    @ASCII
3F3A 4A 36       562         DW    @F<
3F3C 68 36       563         DW    @=0
3F3E             564 ;
3F3E 84 36       565         DW    @PRINT1
3F40 95 36       566         DW    @PRINT2
3F42 9B 36       567         DW    @PRF
3F44 B0 36       568         DW    @PRF2
3F46             569 ;
3F46 C3 36       570         DW    @STRW
3F48 D5 36       571         DW    @STRL
3F4A DD 36       572         DW    @HEXL
3F4C EA 36       573         DW    @VAL1
3F4E 05 37       574         DW    @VAL2
3F50 1A 37       575         DW    @INP$
3F52 3E 37       576         DW    @TRANS1
3F54 47 37       577         DW    @TRANS2
3F56 50 37       578         DW    @FILL
3F58 5E 37       579         DW    @COPYL
3F5A 68 37       580         DW    @DROPL
3F5C 6E 37       581         DW    @SWAPD
3F5E             582 ;
3F5E AB 38       583         DW    @INIT
3F60 DC 38       584         DW    @COL
3F62 F0 38       585         DW    @CLS
3F64 13 39       586         DW    @PALET
3F66 43 39       587         DW    @WIND
3F68 4E 39       588         DW    @LINE
3F6A 5C 39       589         DW    @SLINE
3F6C 67 39       590         DW    @BOX
3F6E 72 39       591         DW    @TILE
3F70 9F 39       592         DW    @TRIANGLE
3F72 86 39       593         DW    @BOXFUL
3F74 AA 39       594         DW    @CIRCLE
3F76 CB 39       595         DW    @POINT
3F78 B5 39       596         DW    @DOT
3F7A C4 39       597         DW    @MAGIC
3F7C             598 ;
3F7C 84 37       599         DW    @PRON
3F7E 87 37       600         DW    @PROFF
3F80 8A 37       601         DW    @PEEK#
3F82 96 37       602         DW    @POKE#
3F84 52 32       603         DW    @END
3F86 61 32       604         DW    @I?
3F88 70 32       605         DW    @J?
3F8A A1 31       606         DW    @CR
3F8C E9 30       607         DW    @ROT
3F8E             608 ;
3F8E 77 32       609         DW    @TR
3F90 8D 32       610         DW    @FR
3F92 AC 32       611         DW    @LEA
3F94 02 36       612         DW    @CMP2
3F96             613 ;
3F96 1D 35       614         DW    @LDIVMD
3F98 6D 35       615         DW    @DDIVMOD
3F9A E8 35       616         DW    @MLT!
3F9C 76 36       617         DW    @INC#
3F9E 7D 36       618         DW    @DEC#
3FA0 10 37       619         DW    @VAL$
3FA2 7D 37       620         DW    @BREAK
3FA4 16 34       621         DW    @MID$
3FA6             622 ; RESERVE
3FA6 00 00       623         DW    0
3FA8 00 00       624         DW    0
3FAA 00 00       625         DW    0
3FAC 00 00       626         DW    0
3FAE             627 ;
```

▶「ワールドスタジアム」が出るのを首を長くして待っているのは私だけでしょうか？
大阪ではプロ野球シーズンは6月まで終わっています。それ以後の野球の話はすべて「ワースタ」内の話です。だから，タイガースがめちゃ強かったりします。

田丸　泰彦(28)大阪府

ようこそここへC言語

［第3回］

制御構造って何だろう

Nakamori Akira
中森 章

制御構造というのはなにやら難しそうな用語ですが，要するにプログラムの流れを表現するものです。C言語には，条件分岐や繰り返しといった処理の手順を記述するために豊富な制御構造が用意されているのです。順を追って見ていきましょう。

始めたらやめらない悪魔のゲーム「シムシティー」をなんとか精神力で封印してこの記事の原稿を書いている中森章です。個人的に「ポピュラス」はいまいちだったのですが，これには当分の間のめり込んでしまいそうな予感がします。

さて，今回のテーマは制御構造です。簡単な表現でいうと選択と繰り返しです。かつて構造化プログラミングの父ダイクストラはアルゴリズムは連接，選択，繰り返しの3つの制御構造で記述できるといいました。裏を返せば，最低限いくつかの制御構造がなければアルゴリズムの記述（＝プログラム）はできません。プログラムとはアルゴリズムそのものなのです[1]。今回はこのありがたい制御構造について学ぶことにしましょう。

---

[1]そういえば昔『データ構造＋アルゴリズム＝プログラミング』という名著があった（編集部注：今でもある）。

制御構造の種類

プログラムには処理の流れというものがあります。それは，まずこれをして，次にあれをして，その次にどれをしてというような処理を行う順序のことです。処理の流れを考えると，通常はある処理が逐次的に実行されるだけですが，ときにはいくつかの選択肢の中からひとつの処理を選んだり，ある処理を何回か繰り返したりすることも必要になります。たとえば，1から与えられた数までの自然数を考えて，その数が奇数ならば加算し，偶数ならば減算した場合の合計値を求めるプログラムを考えてみましょう。これは，数式で表せば，

1－2＋3－4＋5－6＋7－……

の値を求めることです。このプログラムの処理手順は次のようになります。

1)　考える自然数の最大値を変数maxに入力する。
2)　合計値を保持する変数をsumとしてそれに0を代入する（初期化)。
3)　加算または減算する自然数の値を保持する変数をnumberとしてそれに1を代入する（初期化)。
4)　変数numberの値が変数maxの値を越えない限り，5)から6)の処理を繰り返す。
5)　変数numberの値を2で割った余りが1（奇数）ならば変数sumの値に変数numberの値を加える。そうでなければ（偶数）変数sumの値から変数numberの値を引く。
6)　変数numberの値を1だけ増やす。
7)　変数sumの値が求める値になっている。

この処理手順において，5)の処理では加算か減算かという2つの処理からひとつの処理の選択を行い，また，5)から6)の処理が4)の処理の条件のもとで繰り返されています。

このようにちょっとしたプログラムの処理手順を考えただけでも選択や繰り返しという処理が必要になってくるのです。実際，選択や繰り返しという処理なしにはプログラムを書くことはできません。そして，このような選択とか繰り返しといったプログラムの処理の逐次的な流れを変更する仕組みをプログラミング用語で制御構造（control structure）と呼んでいます。

制御構造は大きく次の4種類に分類できます。

●選択制御構造
●繰り返し制御構造
●分岐制御構造
●割り込み処理

これらの処理について簡単に説明しておきましょう。

選択制御構造とは次の3種類の構造の総称です。すなわち，ある条件があって，その条件に合致したとき（あるいは合致しないとき）のみある処理を行う構造。第2に，ある条件に合致したときある処理を行い，合致しないとき別の処理を行う構造。最後に，いくつかの結果を取り得る条件の評価結果にしたがって，それぞれ異なった処理を行う構造です。これらの選択制御構造は時と場合によって微妙に使い分けられます。ただし，すべてのプログラミング言語がこの選択制御構造をすべて備えているわけではありません（C言語にはすべてある)。

繰り返し制御構造は読んで字のごとく，ある処理を繰り返すための構造です。たいていのプログラミング言語には，変数をループカウンタとして使用する繰り返し回数指定型の繰り返し制御構造と，繰り返し処理を終了するための条件を指定する条件指定型の繰り返し制御構造の2種類が備えられています。また，繰り返しの条件判断の時期もループの先頭で行う場合とループの終わりで行うものがあります。

分岐制御構造とは，それまでの処理の流れと直接関係のない部分に処理を変更する制御構造です。BASICなどでお馴染みのGOTO文はこの分岐制御構造に属します。昔からよく議論されるように，分岐を多用するとプログラムの処理があっちに行ったりこっちに来たりで非常に読みにくくなることがあります。このように処理の流れが複雑になったプログラムは「スパゲッティ」と呼ばれ，

プログラムを読む人からは非常に恐れられています。プログラムを書く人ならば一度は耳にするGOTO文廃止論はこのような理由からきているのです。ただし，現在では，GOTO文がまったく不要というわけではなく，エラー発生時にエラー処理に素早く制御を移すためにGOTO文が有用という認識が一般的です。

割り込み処理とはいつ発生するかわからない出来事に対して処理をするための制御構造です。たとえば，実行中のプログラムを停止させたいとき私たちはCTRL＋Cキーを押します。このキーが押されたときプログラムは中断するのですがそれがいつ押されるかはプログラム側では知ることができません。このような不意の出来事に対処するための処理が割り込み処理なのです。ただし，プログラムの処理によってはいきなり中断されると困ることがあります。たとえば，プログラムの開始時に変更した画面モードやパレットを元に戻すことが必要になるかもしれません。

さて，このようにいくつかある制御構造ですが，最初のうちはすべてをマスターする必要はありません。経験上，通常のプログラムで必要なのは選択と繰り返しだけです。

かつて，構造化プログラミングの提唱者で知られるダイクストラ（E.W.Dijkstra）はその著書『構造化プログラミング』（邦訳：サイエンス社）の中で，プログラミングは連接（concatenation），選択（selection），繰り返し（repetition）だけで記述できると述べています[2]。連接とは通常の逐次的処理のことですから，まさに選択と繰り返しの制御構造こそがプログラミングのすべてだと述べているのです。

実際のところ私自身の書いたプログラムを思い出してみても，割り込みはほとんど使用していませんし，分岐に至ってはほんの1回か2回使用したことがあるだけです。初心者は選択と繰り返しさえマスターしておけば大丈夫でしょう。

整理のために，いくつかの基本的な選択制御構造と繰り返し制御構造の処理を流れ図にして図1に示しておきます。

[2]正確には，ダイクストラは理解しやすいプログラムを書くためには連接，選択，繰り返しのみを用いるのがよいということを述べている。その根拠はそれだけの制御構造で十分という考えがあるのだろう。

C言語の制御構造

先に述べた連接および4種類の制御構造のすべてをC言語では使用することができます。図2に制御構造とC言語で使用するための文（や関数）の対応を示しておきます。ただし，今回は図2のすべての文（や関数）を取り上げるのではなく，初心者に必要な連接および選択と繰り返しに焦点を絞って説明します。分岐と割り込みはまた別の機会に譲ります。

●C言語の連接構造

説明するまでもなく，連接とは複数の文の逐次的処理です。ここでは文とは何かということを明確にしておきましょう。C言語における文とは「式のあとにセミコロン（；）を付けたもの」として定義されています。たとえばx＝x＋1とかprintf(“Hello¥n”)といった式（関数呼び出しも式の一種）はセミコロンを付けて，

```
x=x+1;
printf("Hello¥n");
```

とすることで文として認識されるようになります。注意しなければならないのはセミコロンの扱いです。PASCALではセミコロンは文と文の区切りを表す役割を持っていましたが，C言語ではある式が単なる「式」であるか「文」であるかを区別するための識別子なのです。

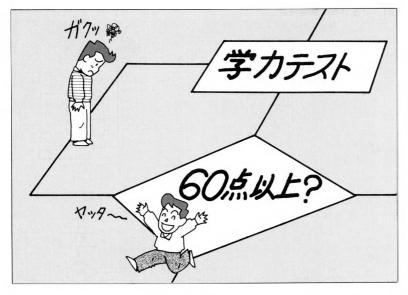
C言語ではいくつかの文をひとまとまりにして単一の文と同等に扱うことができます。これが複文またはブロックと呼ばれる構造です。複文は複数の文を波カッコ｛と｝で囲んだものです。たとえば，

```
{ x=x+1; printf("Hello¥n"); }
```

が複文です。複文の終わりにはセミコロンは付きません。複文は文法的にひとつの文しか記述できない場所に複数

図1　いろいろな制御構造

図2　C言語で使用できる制御構造

制御構造	C言語の文（関数）
選択制御構造	if else switch case default
繰り返し制御構造	while for do while
分岐制御構造	goto return setjmp longjmp (break) (continue)
割り込み処理	signal raise

の文を書くために用いられます。なお，複文は単一の文と同じ扱いですから，複文に含まれる文はそれ自体が複文でもよく，

{ x=x+1； { y=x；z=x*x；} x=y+z；}

などという文も文法上は可能です（複文にはセミコロンが付かないことに注意しよう）。

ところで，C言語には，

;

というようにセミコロンだけからなり，何の処理も行わない空文というものがあります。空文は繰り返し制御構造の処理部で積極的に空ループを作る場合などに用います。初心者が空文を使用することはあまりないと思いますが，他人のプログラムを読むための基礎知識として覚えておきましょう。同様の考えで，

{}

という空ブロックもあります。

●C言語の選択制御構造

C言語の選択制御構造は通常if文とswitch文と呼ばれる構造です。C言語では図1の選択制御構造に示す3種類の構造をすべて備えていて，それぞれは文法的には次のような形式で使用されます。

1） 選択1

・if (exp) stmt;
　　式expの値が0でないなら[3]文stmt[4]を実行し，式expの値が0なら何もしない。
　　(stmt=statement)

・if (exp) { stmt1； stmt2；……}
　　式expの値が0でないなら文stmt1, stmt2, ……を実行し，式expの値が0なら何もしない。

2） 選択2

・if (exp) stmt1 ； else stmt2 ；
　　式expの値が0でないなら文stmt1を実行し，式expの値が0なら文stmt2を実行する。

・if (exp) { stmt11； stmt12；……}
　else stmt21；
　　式expの値が0でないなら文stmt11，stmt12，……を実行し，式expの値が0なら文stmt21を実行する。

・if (exp) stmt1；
　else { stmt21； stmt22；……}
　　式expの値が0でないなら文stmt1を実行し，式expの値が0なら文stmt21，stmt22……を実行する。

・if (exp) { stmt11； stmt12；……}
　else { stmt21； stmt22；……}
　　式expの値が0でないなら文stmt11，stmt12，……を実行し，式expの値が0なら文stmt21，stmt22……を実行する。

3） 選択3

```
・switch ( exp ) {
    case val1  :  stmt11；  stmt12；……
    case val2  :  stmt21；  stmt22；……
                   ⋮
    case valn : stmtn1；  stmtn2；……
  }
```

　　式expの値がval1ならばstmt11，stmt12……を，
　　式expの値がval2ならばstmt21，stmt22……を，
　　⋮
　　式expの値がvalnならばstmtn1，stmtn2……を実行する[5]。

```
・switch ( exp ) {
    case val1  :  stmt11；  stmt12；……
    case val2  :  stmt21；  stmt22；……
                   ⋮
    case valn : stmtn1；  stmtn2；……
    default : stmt1；  stmt2；……
  }
```

　　式expの値がval1ならばstmt11，stmt12……を，
　　式expの値がval2ならばstmt21；stmt22……を，
　　⋮
　　式expの値がvalnならばstmtn1，stmtn2……を実行するが，式expの値がval1からvalnのどれとも一致しないならstmt1，stmt2……を実行する。

●C言語の繰り返し制御構造

C言語での繰り返し制御構造にはwhile文，for文，do文があります。最初に文法を説明しましょう。

1） while文

・while (exp) stmt;
　　式expの値が0でない間，文stmtを繰り返す。

・while (exp) { stmt1； stmt2；……}
　　式expの値が0でない間，文stmt1, stmt2……を繰り返す。

2） for文

・for (exp1； exp2； exp3) stmt;
　　まず，式（文）exp1を実行し，式exp2の値が0でない間，文stmtおよび式（文）exp3の実行を繰り返す。

・for (exp1； exp2； exp3) {stmt1；stmt2；……}
　　まず，式（文）exp1を実行し，式exp2の値が0でない間，文stmt1，stmt2……および式（文）

exp3の実行を繰り返す。

3）　do文

・do stmt; while (exp);

まず文stmtを実行し，式expの値が0でない間，文stmtを繰り返す。

・do { stmt1； stmt2；……} while (exp);

まず，文stmt1，stmt2，……を実行し，式expの値が0でない間，文stmt1，stmt2……を繰り返す。

先に繰り返し制御の種類として，繰り返し回数指定型と終了条件指定型があると説明しましたが，基本的にはfor文が繰り返し回数指定型，while文とdo文が終了条件指定型になります。while文とdo文の違いは繰り返し処理を一度も実行しないことがあるか，最低1回は実行するかの違いです。とはいってもこれらのfor文の形式を見てもどこが繰り返し回数指定型なのかわからないと思います。どう見ても終了条件指定型ですね。これはfor文の使用例を見ないとわかりません。for文は次のような形式で使用されることがほとんどです。

```
for(i=0； i<100； i=i+1) {
  変数 i を使用する処理
}
```

これは，最初に変数 i の値を0に初期化(i=0)し，変数 i の値が100より小さい(i<100)間，変数 i を使用する処理（別に変数 i を使用しなくてもよいが）および変数 i の更新(i=i+1)を繰り返すという記述です。これは繰り返し条件指定型ですね。つまり，for文は応用上は繰り返し回数指定型なのですが，C言語ではもっと一般性を持たせて終了条件指定型にしてあるのです。

C言語の繰り返し制御構造は，繰り返し回数の指定，終了条件の指定といった使い分けらしきものはありますが，結局はどれも終了条件指定型の繰り返しです。このため，すべての制御構造はwhile文で構成することができます。たとえばfor文とdo文はwhile文を用いて次のように変換することもできます。

```
for ( exp1； exp2； exp3 ) stmt;
  → exp1； while ( exp2 ) { stmt; exp3； }
do stmt; while ( exp );
  → stmt; while ( exp ) stmt;
```

このように，C言語の繰り返し制御構造は効能としてはどれも似たようなものですから，それぞれの使用頻度は人によってかなり偏っています。たとえば，for文やdo文をまったく使わずにwhile文だけでプログラムを書く人もいます。傾向としては繰り返し回数が決まっているときはfor文を使い，それ以外はwhile文かdo文の一方のみを使用するという人が多いようです。

●必殺技，breakとcontinue

一般に繰り返し制御構造というものはループの先頭または最後で終了条件の判定が行われます。ところがある場合にはループの途中でループを終了したくなったり，以降の処理を無視したくなることがあります。そのための制御構造がC言語には用意されています。それがbreak文とcontinue文です。

break文は，そのbreak文を囲む最小のswitch文か，繰り返し制御構造（for文，while文，do文）の最小の繰り返し単位の実行を終了させるための文です。break文はその性格上switch文の本体か繰り返し制御構造の本体の中でしか使用することができません。図3にbreak文を使用する場合のswitch文や繰り返し制御構造の制御の流れを示します。

ところで，ここで「最小の」といっている意味は制御構造が入れ子になっている場合，ひとつだけ外側にのみ抜け出すことができるということです。たとえば，

```
while(i<1000) {
  n=n*i；
  for(j=0；j<2000；j=j+1) {
    if (j>n) break；
    n=n+j；
  }
  i=i*i；
}
```

という入れ子の繰り返し構造を考えましょう（このプログラムはまったく無意味ですから何をやっているのか考えるのはやめましょう）。if文の条件が成立してbreak文

図3　break文の制御の流れ

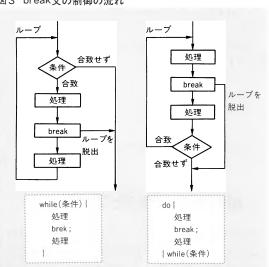
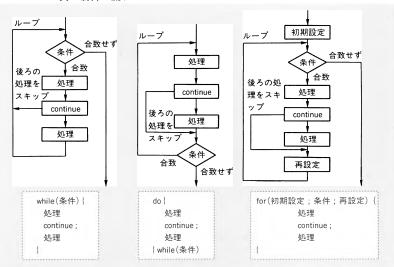



図4　continue文の制御の流れ

が実行されると，内側のループであるfor文は終了してしまいますが，外側のループであるwhile文は影響を受けません。

一方continue文は，continue文を含む最小の繰り返し制御構造のループの終わりまで処理をスキップするための文です。continue文は繰り返し制御構造の中でcontinue文以降の文を無視して次の繰り返しに移りたいときに使用します。continue文はbreak文とは違い，switch文の中に書くことはできません。図4にcontinue文を使用する場合の繰り返し制御構造の制御の流れを示します。

continue文はぜひとも必要な文というわけではありません。これはelse付きのif文や字下げを多用することによってプログラムが見にくくなるのを防ぐために使用します。たとえば，

```
while(i<1000) {
  i=i*i;
  if(i<100) {
      簡単な処理
  }
  else {
      複雑な処理
  }
}
```

リスト1　while文を使用したプログラム

```
 1: /*
 2:      リスト1    ***  while を使ったプログラム ***
 3: */
 4:
 5: main()
 6: {
 7:         int max,sum,number;
 8:
 9:         printf("最大値?");
10:         scanf("%d", &max);
11:
12:         sum = 0;
13:         number = 1;
14:         while( number <= max ){
15:                 if( (number % 2)==1 ) /* 2で割った余りが1 → 奇数 */
16:                         sum = sum+number;
17:                 else                  /* 2で割った余りが0 → 偶数 */
18:                         sum = sum-number;
19:                 number = number+1;
20:         }
21:
22:         printf("結果=%d¥n", sum);
23: }
```

リスト2　for文を使用したプログラム

```
 1: /*
 2:      リスト2    ***  for を使ったプログラム ***
 3: */
 4:
 5: main()
 6: {
 7:         int max,sum,number;
 8:
 9:         printf("最大値?");
10:         scanf("%d", &max);
11:
12:         sum = 0;
13:         for ( number = 1; number <= max; number=number+1 ){
14:                 switch ( number % 2 ){
15:                 case 1:
16:                         sum = sum+number;
17:                         break;
18:                 case 0:
19:                         sum = sum-number;
20:                         break;
21:                 }
22:         }
23:
24:         printf("結果=%d¥n", sum);
25: }
```

というプログラムがある場合，continue文を使って，

```
while(i<1000) {
  i=i*i;
  if(i<100) {
      簡単な処理
      continue;
  }
  複雑な処理
}
```

と書けば，else文による字下げを省略でき，プログラムがスッキリとします。

3) C言語で条件を示す場合，0が偽を，0以外が真を示す。if文などの条件指定としては，x<=yなどという条件式(関係式)が普通であるが，x+yなどの一般の(整数を値とする)式も使用できる。C言語の条件式は，指定した関係が満たされる場合は1，それ以外の場合は0が値となる。

4) 正確には文ではなく式stmtと書くべきである。if, else, do, whileなどの処理部には単一の文または単一の複文しか記述できないのでセミコロンが付いた文では間違いである（これでは文+空文の2文である)。しかし，この連載では視覚的な理解を第一として，ひとつの文をあえてstmt;と表現している。

5) 実はこの説明は正しくない(こればっか)。switch文では条件式に一致する値を持つcaseラベル以降のすべてのcaseラベル（defaultを含む）の処理を順次実行する（caseの突き抜け)。このため，条件式の値ごとに処理を切り分けたいのであれば，ひとつのcaseラベルの処理が終了したあとにswitch文から抜け出すための明示的な指定をしなければならない。そのための指定はbreak文またはreturn文で行う。C言語のswitch文がこのような仕様になっている理由は多重のcaseラベルに対して単一の処理を記述できるようにするためである。それ以外の目的での使用は健全でないとK&Rにも書かれている。

◆基礎力を高めよう

設問1　C言語ではいくつかの文を｛と｝で囲んで複文を作り，PASCALではいくつかの文をbeginとendで囲んで複文を作ります。複文内におけるセミコロンの扱いの違いを例を挙げて説明してください。

設問2　次のif文の解釈として正しいのは，それぞれA)，B)のうちどちらでしょう。

```
if ( E1 ) if ( E2 ) S1; else S2;
→
A) if ( E1 ) {
     if ( E2 ) S1; else S2;
   }
   ※elseはif(E2)に対応する。
B) if ( E1 ) {
     if ( E2 ) S1;
   } else {
     S2;
   }
   ※elseはif(E1)に対応する。
```

(解答は138ページ)

制御構造を用いたプログラム

C言語の制御構造について主なものをひと通り説明してきました。ここでは学んだ制御構造を使って実際のプログラムを作ってみましょう。例題として，今回の制御

構造の説明の最初に挙げた，

$$1-2+3-4+5-\cdots\cdots$$

を求めるプログラムをいろいろな制御構造を用いて書いてみましょう。プログラムの形式は先月号で示したテンプレート（定型的な枠組み）にあてはめ，scanf関数で自然数の最大値を入力し，求める値を計算したあと，printf関数で計算結果を出力するものとします。リスト1，リスト2，リスト3が，それぞれwhile文，for文，do文を使用したプログラムです。興味のある人は実行してみてください。

なお，リスト1とリスト3では，加算（または減算）する自然数が偶数であるか奇数であるかの判断はif文で行っていますが，リスト2ではswitch文を使ってみました。ひとつのアルゴリズムを実現するための方法はひとつではないということを実感してください。

ところで，リスト1からリスト3のプログラムはあまり効率がよくありません。

$$1-2+3-4+5-\cdots\cdots$$

という式を見たとき，

$$\begin{aligned}&1-2+3-4+\cdots\cdots+(2n)\\&=(1-2)+(3-4)+\cdots\cdots+[(2n-1)-(2n)]\\&=(-1)+(-1)+\cdots\cdots+(-1)\\&=-n\end{aligned}$$

$$\begin{aligned}&1-2+3-4+\cdots\cdots+(2n+1)\\&=1+(-2+3)+(-4+5)+\cdots\cdots+[-(2n)+(2n+1)]\\&=1+1+\cdots\cdots+1\\&=n+1\end{aligned}$$

を発見すればリスト4のようなプログラムを書くこともできます。リスト1からリスト3のプログラムは与える自然数の上限値が大きくなるにつれて計算時間が増えてきますが，リスト4ではどれも同じ時間で計算できてしまいます[6)]。

ときには問題をよく吟味してプログラムを書くことも必要ですね。と教訓を得たところで今回はこのくらいにしておきましょう。

---

6)天才数学者のガウスは少年時代，1から与えられた数値までの和を即座に答えて彼の先生を驚かせたという。ガウスはすでに等差数列の和の公式を知っていたわけだ。このように本筋とは違う方法で効率のよいプログラムを書くとガウスのように偉くなったような気分になれるかも。

今月はここまで

今回制御構造を覚えることで逐次的処理しかやれなかったC言語のプログラムにも自由度が増えてきました。これでやっとC言語によるプログラミングの入り口にさしかかったことになります。C言語で本当に「使える」プログラムを作るためにはまだまだ覚えなければならないことがたくさんありますが，プログラミングの中で今回の制御構造がもっとも大切なものです。各自，制御構造を応用したいろいろなプログラムを書いてみて，使い方を着実に身につけるようにしてください。来月は配列と文字列について説明したいと思います。それではまたお会いしましょう。

◆参考文献

C compiler PRO-68K各マニュアル，シャープ

プログラミング言語C第二版，共立出版

◆基礎力を高めようの解答

設問1　セミコロンはC言語では文の終わり，PASCALでは文と文の区切りを表す。したがって，C言語では（複文でない）文の終わりには必ずセミコロンが必要であるが，PASCALでは文が連続していない場合はセミコロンは必要ない。この違いは複文内の最後の文に現れる。PASCALでは複文内の最後の文，すなわちendの直前の文の最後にセミコロンを付けても付けなくてもよい。しかし，C言語では }の直前の文の最後には（それが複文でなければ）必ずセミコロンが必要である。

すなわち，PASCALでは，

```
begin x:=x+1 ; y:=y+1 ; end
begin x:=x+1 ; y:=y+1 end
```

はどちらも正しい複文である。

C言語では，

```
{ x=x+1 ; y=y+1 ; }
```

は正しいが，

```
{ x=x+1 ; y=y+1 }
```

は正しくない。

設問2　A)が正しい。elseはそれより前の一番近いifに対応する。同じような考え方で，

```
if (E1) S1 ; else if (E2) S2 ; else S3 ;
```

は，

```
if (E1) S1 ; else { if (E2) S2 ; else S3 ; }
```

であることがわかる。

リスト3　do文を使用したプログラム

```
 1: /*
 2:      リスト3    *** do ～ while を使ったプログラム ***
 3: */
 4:
 5: main()
 6: {
 7:         int max,sum,number;
 8:
 9:         printf("最大値?");
10:         scanf("%d", &max);
11:
12:         sum = 0;
13:         number = 1;
14:         do {
15:                 if( (number % 2)==1 ) /* 奇数 */
16:                         sum = sum+number;
17:                 else                  /* 偶数 */
18:                         sum = sum-number;
19:                 number = number+1;
20:
21:         } while( number <= max );
22:
23:         printf("結果 = %d¥n", sum);
24: }
```

リスト4　もっとも効率のよいプログラム

```
 1: /*
 2:      リスト4    *** もっとも効率のいいプログラム ***
 3: */
 4:
 5: main()
 6: {
 7:         int max,sum;    /* number の宣言は要らない */
 8:
 9:         printf("最大値?");
10:         scanf("%d", &max);
11:
12:         switch ( max % 2 ){ /* 普通はif文で書く */
13:         case 1:
14:                 sum = max/2 + 1;
15:                 break;
16:         case 0:
17:                 sum = -max/2;
18:                 break;
19:         }
20:
21:         printf("結果 = %d¥n", sum);
22: }
```

マシン語カクテル in Z80's Bar

第17回——私はエディタ,原稿まだかな——

シナリオ＆イラスト：山田純二
特別監修：浦川博之

前回のややこしい雰囲気を引きずりながら進んでいく今回のマシン語カクテル。さて，プログラムはというと山田君がなんの脈絡もなく変な扮装で持ってきてくれるS-OS用のカーソルエディタです。どのようにでも自由にご使用ください。

♪カラン，コロ～ン
マスター（以下M）：いらっしゃい。
メアリー（以下メ）：イラッシャーイ。
長老（以下老）：どうも，ひさしぶりじゃな。あい変わらず静かじゃのう。
M：余計なお世話ですよ，今日はおひとりで？
老：うむ，2，3人若い連中をつれてこようと思ったが，みな忙しそうだったからの。
M：そういえば，常連の山田君も最近顔を見せないな。ああそうだ。ようこちゃん，光君は？
ようこ（以下Yo）：知らないわよ。
老：冷たいもんじゃのう。ところで，ようこちゃん，その手に持ってるものはなにかな。
Yo：この間，マスターが骨董品屋で買ってきた壺よ。
老：ずいぶんとホコリにまみれているし，形も妙な壺じゃな。
M：なんでも，インドのとある寺院に置いてあったものだということですよ。
老：ふ～ん，本当かのう。
Yo：いいじゃない。どうせ安かったんでしょ。
メ：そういうコトデ一ス。汚れたママジャカワイソウだから，キレイにシテアゲマショウ（キュッキュッ）。
老：げほっげほっ。ものすごいホコリじゃ……。ヘックショーイっとな。
壺：モクモクモク～。
M：うわっ，なんだこの煙は。ようこちゃん，消火器！　消火器！
？：ハッハッハ～，ダイジョウブイ，じゃなくて，呼ばれて飛び出てじゃじゃじゃじゃ～ん。
Yo：あっ，謎のアラビア人。
メ：Oh！　違イマ～ス。ハクション大魔王デ～ス。
老：なにをいっておるんじゃ。なんだ，誰かと思えば山田君じゃないか。凝った変装をしおって。
純二（以下純）：そのわりにはあっさりばれてしまいましたね。
M：今日はなにかプログラムを持ってきたんですか。
純：よくぞ聞いてくれました。構想2時間，プログラミングに8時間かけた，テキストエディタを持ってきたんですよ。

エディタとは？

老：ところで，ようこちゃんエディタとはなにか知っているかの。
Yo：知っているわよ。雑誌なんかでライターと呼ばれる人たちを使って原稿を集め，きびしい進行に泣きながら，本を作っている人たちのことでしょう。
純：それは，編集者。
老：たしかに編集者のこともエディタというが，この場合は……。
メ：コンピュータのプログラムヲ書クタメノ道具デスネ。
老：メアリーのいうとおりじゃ。プログラミングをするときに，作成したいプログラムを管理するプログラムのことをいっているのじゃろう。
Yo：プログラムを作るためのプログラム。いまいちピンとこないなあ。
純：たいていの場合，そんなことは気にしないで使っているからね。
M：ところで純二君，今回はどういったタイプのエディタを作ってきたんですか。
純：BASICやZEDAに付属しているカーソルエディタです。
メ：ソレハ，ドウイッタモノデスカ。
老：1行ごとについている行番号によって，それぞれの行をエディットしていくものじゃよ。
Yo：ほかにはどういった種類があるの。
老：ラインエディタ，スクリーンエディタなんかがあるのう。
Yo：いちばん使いやすいものはどれなの。
老：やっぱり，編集するものの中を自由に行ったり来たりして，どこでも編集できるスクリーンエディタじゃな。
Yo：ふーん，だったら純二君，あまり使い勝手のよくないカーソルエディタじゃなくてスクリーンエディタを作ればよかったのに。
純：たしかにそうかもしれませんが，スクリーンエディタはカーソルエディタに比べてはるかに複雑なんですよ。スクリーンエディタを作るとなると，かなりの行数が必要となるけど，カーソルエディタならほんの数百行ですみますから。
メ：ソウソウ，小サイコトハいいコトネ。
M：なんのこっちゃ。

さあ，作るぞ

老：さて，ようこちゃん，編集という作業はどういうものかわかるかな？
Yo：文字を入力すること。
老：ちょっと違うな，編集というのは入力されたものを○○，☆☆したり？？したりすることをいうんじゃよ。
Yo：やっだ～，長老のス・ケ・ベ。
老：な，なにを考えておるのじゃ。冗談いってないで真面目に答えなさい。
純：長老のいけず～う。
老：ばかもの（ポカッ）！　おぬしまで一緒になってなにをいっておるんじゃ。
純：痛いなあ……。わざわざ伏せ字にしていうから変な誤解をまねくんですよ。答えは，削除，追加，更新することでしょう。
老：そういうことじゃ。素直に答えておれ

ば痛い目に遭わずにすんだものを。

メ：冗談ハソレグライニシテ，具体的ナ方法ヲ，教エテクダサーイ。

老：よかろう。まず，テキストがどのような構成となっているかを説明してくれんか，純二君。

純：はい。S-OSの場合，行のエンドコードは$0D_H$で，テキストのエンドコードは00_Hとなっています。

老：うむ，次に必要なワークはなにがあるかな。

純：テキストを格納している先頭アドレス(#TEXTENT)，テキストの終了アドレス(#TEXTEND)，編集している行の先頭アドレス（#EDADR）の３つです。

Yo：たった３つだけでいいの。

老：そうじゃよ，なんとなく難しそうに思えてしまうが，やっていることは思ったより単純なんじゃよ。では，行の削除はどうやっているか，見ていくとするか。

純：行の削除は，編集している行の先頭アドレスへ次の行の先頭アドレスから最終行までのテキストを転送してやれば完了です（図１）。

メ：プログラムではドウナッテイルンデスカ？

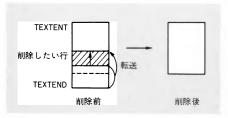
純：それは，

1)　削除したい行の先頭アドレスをDEレジスタに格納する
2)　削除したい次の行の先頭アドレスをHLレジスタに格納する
3)　（#TEXTEND）からHLの値を引いたもの＋1をBCレジスタに格納する
4)　LDIR命令で転送する
5)　DE−1の値を新しい(#TEXTEND)の値として格納する

と，いうふうにやるんですよ。

老：次は，行の追加じゃな。

純：行の追加は，追加したい行から最終行までのテキストを，追加する行の桁数だけ下に転送してから，追加する行の内容をメモリに転送してやればいいのです（図２）。具体的な手順は，

1)　追加したい行の先頭アドレスをHLレジスタに求める
2)　PUSH HLで値を保存
3)　入力された行の桁数をBCレジスタに求める
4)　（#TEXTEND）の値をHLレジスタにロードする
5)　HLレジスタにさっき求めたBCレジスタの値を足したものを，DEレジスタの値とする(これが，新しい(#TEXTEND)の値)
6)　LDDR命令で転送する
7)　POP HLで保存した値を取り出し，そのアドレスに入力された行を転送する

という具合になります。

Yo：よーし，次は行の更新ね。

純：行の更新には２通りの方法があるんです。まずひとつ目は，更新したい行を削除してから，新しく入力された行を追加する方法。

メ：サッキノ削除ト追加ノプログラムをツカエバイイノデスネ。

純：そのとおり。使えるものはできるかぎり共用してしまおうという精神に基づくものです。プログラムも短く収まるので，とってもうれしいのですが，実行速度の点でもうひとつの方法に劣るところがあるんですよ。

Yo：で，その方法って？

善司（以下善）：あるわけないんだよ～ん。

Yo：純二く～ん。嘘をついたわね。

純：うわあ，ようこさんマジになって怒らないでくださいよ。それに善ちゃん，いきなりわいて出てきてとんでもないことをいわないでよ。

善：いやあ，みんなで真面目な顔をしていたから，場をなごませようとしただけですよ。

老：いらんことばっかりに，気が回る男じゃのう。ま，こんなやつはほっといてさっさと説明してくれんか。

純：はい，削除してから追加するとたしかにプログラムの構造は単純となります。もうひとつの方法とは，更新する行と入力された行の長さを比べて，

1)　更新する行＝入力された行
2)　更新する行＞入力された行
3)　更新する行＜入力された行

の３通りに場合分けをしてやるのです。1)の場合にはそのまま入力された行を転送してやり，2)の場合には余った桁を詰め，3)の場合にははみ出した桁の分空けてやればいいのです。この方法だとブロック転送の回数が１回ですむから効率はいいんですよ。まあ，今回は行数をできるだけ短くしたかったので前者の方法を使いましたけどね。

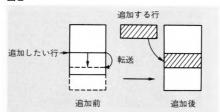



図１

図２

ZED-645の使用法

M：それじゃあ，プログラムの使い方を説明してくれませんか。

純：使えるコマンドはカコミにあるとおりです。使い方についてはZEDAの付属のエディタとまったく同じです。違うのは行の指定に16進４桁の数値を使っている点でしょう。行数の関係上，10進16進変換ルーチンを入れるスペースがなくなってしまったのです。

Yo：コマンドを見てみると削除と追加のコマンドはあるけど，行の更新はどうやっているの。

純：それは，リストの最初のほうを見てください。１行入力が終わったあと，入力された行の先頭がプロンプト“>”であった場合にはコマンド判定ルーチンへ行き，そうでなかった場合には行の更新を行うのです。“T”コマンドでリストを表示してから変更したい行をエディットしていけばいいんですよ。

老：なるほどな，しかし短いだけあって本当に最低限の機能しかついておらんな。

純：まあ，短いというだけあって小回りが利くし，もし，ほしい機能があったなら自分でどんどん拡張していけばいいと思いま

ZED-645コマンド一覧表

nは16進４桁。

Tn……n行からテキストを表示していきます。nを省略した場合は０行目から表示します。

In……n行目からテキストを挿入していきます。このコマンドを実行後はテキスト挿入モードになりますので，抜けたいときにはSHIFT＋BREAKを押してください。nを省略した場合には（#EDADR）の指すアドレスからテキストを挿入していきます。

Dn……n行目のテキストを削除します。nは省略不可能です。

Xaddress……テキスト格納アドレスを16進４桁で指定します。アドレスを省略すると現在のメモリの格納状態が表示されます。

&……テキストをクリアします。

R……テキストを復活させます。

Sfile name……現在編集中のテキストをセーブします。

Lfile name……指定されたファイルネームのテキストをロードします。

Q……エディタを終了してS-OSのシステムに戻ります。

すけどね。それが，このエディタの正しい使い方だと思います。

帰ろ，帰ろ

M：といったところで，そろそろおひらきですが，山田君いきなり登場したりして，今月も光君がやるはずじゃなかったの。

純：彼もなにかと忙しいようで。

メ：ワタシ，サミシイデス。

老：ようこちゃんも，さみしいじゃろう。

Yo：別に。

善：ま，そういうことにしときましょうか。

純：それじゃあ，僕はこれで失礼します。長老と善ちゃん一緒に帰らない？

老：わしはもう少しここにいることにするから，先に帰っていいぞ。

善：僕も，もうちょっと残っていくから。

純：ふーん。じゃあ，またね。

M：ありがとうございました。

♪カラン，コローン

つづく

リスト1

```
0000                 1  ; ZED-645
0000                 2  ;   in Z80's Bar
0000                 3  ;         1990.10.15 by J.YAMADA
0000                 4
A000                 5           ORG     $A000
A000                 6
1FF4 P               7  #PRINT  EQU $1FF4
1FF1 P               8  #PRINTS EQU $1FF1
1FEE P               9  #LTNL   EQU $1FEE
1FE5 P              10  #MSX    EQU $1FE5
1FE8 P              11  #MSG    EQU $1FE8
1FE2 P              12  #MPRINT EQU $1FE2
1FC4 P              13  #BELL   EQU $1FC4
1FC1 P              14  #PRTHX  EQU $1FC1
1FBE P              15  #PRTHL  EQU $1FBE
1FB2 P              16  #HLHEX  EQU $1FB2
1FC7 P              17  #PAUSE  EQU $1FC7
1FD3 P              18  #GETL   EQU $1FD3
1F76 P              19  #KBFAD  EQU $1F76
2018 P              20  #CSR    EQU $2018
201E P              21  #LOC    EQU $201E
1FAF P              22  #WOPEN  EQU $1FAF
1FAC P              23  #WRD    EQU $1FAC
1FA6 P              24  #RDD    EQU $1FA6
1FA3 P              25  #FILE   EQU $1FA3
1FA0 P              26  #FSAME  EQU $1FA0
1F9D P              27  #FPRNT  EQU $1F9D
2009 P              28  #ROPEN  EQU $2009
1F72 P              29  #SIZE   EQU $1F72
1F70 P              30  #DTADR  EQU $1F70
1F6E P              31  #EXADR  EQU $1F6E
2033 P              32  #ERROR  EQU $2033
1F6A P              33  #MEMAX  EQU $1F6A
1F9D P              34  #FPRINT EQU $1F9D
A000                35
A000 18 09          36  COLD     JR      COLD2
A002 18 1D          37  HOT      JR      HOT2
A004                38
A004 00 4E          39  #TEXTENT DW   $4E00
A006 00 4E          40  #TEXTEND DW   $4E00
A008 00 00          41  #EDADR   DW   0000
A00A 00             42  #BACKCHR DB   00
A00B                43
A00B                44  COLD2
A00B CD E2 1F       45           CALL    #MPRINT
A00E 0C 2A 2A 20    46           DB      $0C,"** ZED-645 **",$0D,00
A012 5A 45 44 2D
A016 36 34 35 20
A01A 2A 2A 0D 00
A01E CD 73 A0       47           CALL    #TEXTNEW
A021                48  HOT2
A021 2A 04 A0       49           LD      HL,(#TEXTENT)
A024 22 08 A0       50           LD      (#EDADR),HL
A027 CD 5A A2       51           CALL    #ENDSEARCH
A02A                52
A02A                53  ;COMMAND LINE
A02A                54
A02A                55  COM
A02A 3E 3E          56           LD      A,">"
A02C CD F4 1F       57           CALL    #PRINT
A02F ED 5B 76 1F    58           LD      DE,(#KBFAD)
A033 CD D3 1F       59           CALL    #GETL
A036 1A             60           LD      A,(DE)
A037 21 02 A0       61           LD      HL,HOT
A03A E5             62           PUSH    HL
A03B FE 3E          63           CP      ">"
A03D C2 48 A1       64           JP      NZ,#REWRITE
A040 ED 5B 76 1F    65           LD      DE,(#KBFAD)
A044 13             66           INC     DE
A045 1A             67           LD      A,(DE)
A046 B7             68           OR      A
A047 C8             69           RET     Z
A048 FE 51          70           CP      "Q"
A04A 28 25          71           JR      Z,EXIT
A04C FE 26          72           CP      "&"
A04E 28 23          73           JR      Z,#TEXTNEW
A050 FE 54          74           CP      "T"
A052 28 59          75           JR      Z,#LISTPRT
A054 FE 52          76           CP      "R"
A056 28 28          77           JR      Z,#RECOVER
A058 FE 44          78           CP      "D"
A05A CA E1 A0       79           JP      Z,#DELETE
A05D FE 58          80           CP      "X"
A05F 28 2A          81           JR      Z,#ADRSET
A061 FE 49          82           CP      "I"
A063 CA FF A0       83           JP      Z,#INSERT
A066 FE 53          84           CP      "S"
A068 CA 13 A2       85           JP      Z,#SAVE
A06B FE 4C          86           CP      "L"
A06D CA BE A1       87           JP      Z,#LOAD
A070 C9             88           RET
A071                89  EXIT
A071 E1             90           POP     HL
A072 C9             91           RET
A073                92
A073                93  ;TEXT CLEAR
A073                94
A073                95  #TEXTNEW
A073 2A 04 A0       96           LD      HL,(#TEXTENT)
A076 7E             97           LD      A,(HL)
A077 32 0A A0       98           LD      (#BACKCHR),A
A07A 36 00          99           LD      (HL),00
A07C 22 06 A0      100           LD      (#TEXTEND),HL
A07F C9            101           RET
A080               102
A080               103  ;TEXT RCOVER
A080               104
A080               105  #RECOVER
A080 3A 0A A0      106           LD      A,(#BACKCHR)
A083 2A 04 A0      107           LD      HL,(#TEXTENT)
A086 77            108           LD      (HL),A
A087 CD 5A A2      109           CALL    #ENDSEARCH
A08A C9            110           RET
A08B               111
A08B               112  ;TEXT KAKUNOU ADRESS
A08B               113
A08B               114  #ADRSET
A08B 13            115           INC     DE
A08C CD B2 1F      116           CALL    #HLHEX
A08F 30 15         117           JR      NC,ADS2
A091 2A 04 A0      118           LD      HL,(#TEXTENT)
A094 CD BE 1F      119           CALL    #PRTHL
A097 3E 2D         120           LD      A,"-"
A099 CD F4 1F      121           CALL    #PRINT
A09C 2A 06 A0      122           LD      HL,(#TEXTEND)
A09F CD BE 1F      123           CALL    #PRTHL
A0A2 CD EE 1F      124           CALL    #LTNL
A0A5 C9            125           RET
A0A6               126  ADS2
A0A6 22 04 A0      127           LD      (#TEXTENT),HL
A0A9 CD 73 A0      128           CALL    #TEXTNEW
A0AC C9            129           RET
A0AD               130
A0AD               131  ;LIST PRINT
A0AD               132
A0AD               133  #LISTPRT
A0AD 13            134           INC     DE
A0AE CD B2 1F      135           CALL    #HLHEX
A0B1 38 09         136           JR      C,LPRT5
A0B3 5D            137           LD      E,L
A0B4 54            138           LD      D,H
A0B5 D5            139           PUSH    DE
A0B6 CD 76 A1      140           CALL    #EDADRSET
A0B9 D1            141           POP     DE
A0BA 18 06         142           JR      LPRT4
A0BC               143  LPRT5
A0BC 2A 04 A0      144           LD      HL,(#TEXTENT)
A0BF 11 00 00      145           LD      DE,0000
A0C2               146  LPRT4
A0C2 7E            147           LD      A,(HL)
A0C3 B7            148           OR      A
A0C4 C8            149           RET     Z
A0C5               150  LPRT2
A0C5 EB            151           EX      DE,HL
A0C6 CD BE 1F      152           CALL    #PRTHL
A0C9 3E 3A         153           LD      A,":"
A0CB CD F4 1F      154           CALL    #PRINT
A0CE CD E8 1F      155           CALL    #MSG
A0D1 CD EE 1F      156           CALL    #LTNL
A0D4 CD C7 1F      157           CALL    #PAUSE
A0D7 E0 A0         158           DW      LPRT3
A0D9 EB            159           EX      DE,HL
A0DA 13            160           INC     DE
A0DB CD B0 A1      161           CALL    #INCADR
A0DE 20 E5         162           JR      NZ,LPRT2
A0E0 C9            163  LPRT3    RET
A0E1               164
A0E1               165  ;TEXT 1 LINE DELETE
A0E1               166
A0E1               167  #DELETE
A0E1 13            168           INC     DE
A0E2 CD B2 1F      169           CALL    #HLHEX
A0E5 38 14         170           JR      C,DELERR
A0E7 CD 76 A1      171           CALL    #EDADRSET
A0EA 38 0F         172           JR      C,DELERR
A0EC               173  #DELSUB
A0EC 5D            174           LD      E,L
A0ED 54            175           LD      D,H
A0EE CD B0 A1      176           CALL    #INCADR
A0F1 CD 8A A1      177           CALL    #SIZESET
A0F4 ED B0         178           LDIR
A0F6 1B            179           DEC     DE
A0F7 22 06 A0      180           LD      (#TEXTEND),HL
A0FA C9            181           RET
A0FB               182  DELERR
A0FB CD C4 1F      183           CALL    #BELL
A0FE C9            184           RET
A0FF               185
A0FF               186  ;TEXT INSERT
A0FF               187
A0FF               188  #INSERT
A0FF 13            189           INC     DE
A100 CD B2 1F      190           CALL    #HLHEX
A103 30 05         191           JR      NC,IS2
```

```
A105 2A 08 A0      192          LD      HL,(#EDADR)
A108 18 03         193          JR      IS4
A10A               194  IS2
A10A CD 76 A1      195          CALL    #EDADRSET
A10D               196  IS4
A10D ED 5B 76 1F   197          LD      DE,(#KBFAD)
A111 CD D3 1F      198          CALL    #GETL
A114 1A            199          LD      A,(DE)
A115 FE 1B         200          CP      $1B
A117 C8            201          RET     Z
A118 CD 2D A1      202          CALL    #INSSUB
A11B ED 5B 76 1F   203          LD      DE,(#KBFAD)
A11F CD A4 A1      204          CALL    #TRNS
A122 2A 08 A0      205          LD      HL,(#EDADR)
A125 CD B0 A1      206          CALL    #INCADR
A128 22 08 A0      207          LD      (#EDADR),HL
A12B 18 E0         208          JR      IS4
A12D               209  #INSSUB
A12D E5            210          PUSH    HL
A12E CD 8A A1      211          CALL    #SIZESET
A131 C5            212          PUSH    BC
A132 EB            213          EX      DE,HL
A133 CD 98 A1      214          CALL    #LENGTH
A136 06 00         215          LD      B,00
A138 2A 06 A0      216          LD      HL,(#TEXTEND)
A13B 5D            217          LD      E,L
A13C 54            218          LD      D,H
A13D EB            219          EX      DE,HL
A13E 09            220          ADD     HL,BC
A13F 22 06 A0      221          LD      (#TEXTEND),HL
A142 EB            222          EX      DE,HL
A143 C1            223          POP     BC
A144 ED B8         224          LDDR
A146 E1            225          POP     HL
A147 C9            226          RET
A148               227
A148               228  ;TEXT REWRITE
A148               229
A148               230  #REWRITE
A148 CD B2 1F      231          CALL    #HLHEX
A14B D8            232          RET     C
A14C 13            233          INC     DE
A14D D5            234          PUSH    DE
A14E CD 76 A1      235          CALL    #EDADRSET
A151 D1            236          POP     DE
A152 D5            237          PUSH    DE
A153 D5            238          PUSH    DE
A154 E5            239          PUSH    HL
A155 CD EC A0      240          CALL    #DELSUB
A158 E1            241          POP     HL
A159 D1            242          POP     DE
A15A CD 2D A1      243          CALL    #INSSUB
A15D D1            244          POP     DE
A15E CD A4 A1      245          CALL    #TRNS
A161 CD 18 20      246          CALL    #CSR
A164 2E 05         247          LD      L,05
A166 CD 1E 20      248          CALL    #LOC
A169 ED 5B 76 1F   249          LD      DE,(#KBFAD)
A16D CD D3 1F      250          CALL    #GETL
A170 1A            251          LD      A,(DE)
A171 FE 1B         252          CP      $1B
A173 C8            253          RET     Z
A174 18 D2         254          JR      #REWRITE
A176               255
A176               256  ; EDIT POINTER SET
A176               257  ; IN HL=EDIT LINE NO.
A176               258
A176               259  #EDADRSET
A176 EB            260          EX      DE,HL
A177 2A 04 A0      261          LD      HL,(#TEXTENT)
A17A               262  EDS2
A17A 7A            263          LD      A,D
A17B B3            264          OR      E
A17C 28 08         265          JR      Z,EDS3
A17E CD B0 A1      266          CALL    #INCADR
A181 38 03         267          JR      C,EDS3
A183 1B            268          DEC     DE
A184 18 F4         269          JR      EDS2
A186               270  EDS3
A186 22 08 A0      271          LD      (#EDADR),HL
A189 C9            272          RET
A18A               273
A18A               274  ;(TEXTEND)-HL
A18A               275
A18A               276  #SIZESET
A18A D5            277          PUSH    DE
A18B EB            278          EX      DE,HL
A18C 2A 06 A0      279          LD      HL,(#TEXTEND)
A18F B7            280          OR      A
A190 ED 52         281          SBC     HL,DE
A192 4D            282          LD      C,L
A193 44            283          LD      B,H
A194 03            284          INC     BC
A195 EB            285          EX      DE,HL
A196 D1            286          POP     DE
A197 C9            287          RET
A198               288
A198               289  ;1 LINE LENGTH
A198               290
A198               291  #LENGTH
A198 0E 00         292          LD      C,00
A19A               293  LEN2
A19A 0C            294          INC     C
A19B 7E            295          LD      A,(HL)
A19C B7            296          OR      A
A19D C8            297          RET     Z
A19E FE 0D         298          CP      $0D
A1A0 C8            299          RET     Z
A1A1 23            300          INC     HL
A1A2 18 F6         301          JR      LEN2
A1A4               302
A1A4               303  ;BAFA TO MEMORY
A1A4               304  ;   IN    DE=BAFA ADRESS,HL=KAKUNOU ADRESS
A1A4               305
A1A4               306  #TRNS
A1A4 1A            307          LD      A,(DE)
A1A5 B7            308          OR      A
A1A6 28 05         309          JR      Z,TRS2
A1A8 77            310          LD      (HL),A
A1A9 23            311          INC     HL
A1AA 13            312          INC     DE
A1AB 18 F7         313          JR      #TRNS
A1AD               314  TRS2
A1AD 36 0D         315          LD      (HL),$0D
A1AF C9            316          RET
```

```
A1B0               317
A1B0               318  ;1 LINE ADDRESS FORWARD
A1B0               319
A1B0               320  #INCADR
A1B0 7E            321          LD      A,(HL)
A1B1 B7            322          OR      A
A1B2 28 08         323          JR      Z,#INCA4        ;TEXT END
A1B4 23            324          INC     HL
A1B5 FE 0D         325          CP      $0D
A1B7 20 F7         326          JR      NZ,#INCADR
A1B9 7E            327          LD      A,(HL)
A1BA B7            328          OR      A
A1BB C0            329          RET     NZ
A1BC               330  #INCA4
A1BC 37            331          SCF
A1BD C9            332          RET
A1BE               333
A1BE               334  ;TEXT LOAD
A1BE               335
A1BE               336  #LOAD
A1BE 13            337          INC     DE
A1BF 3E 04         338          LD      A,04
A1C1 CD A3 1F      339          CALL    #FILE
A1C4               340  #LOAD3
A1C4 CD 09 20      341          CALL    #ROPEN
A1C7 38 3C         342          JR      C,LOADERR
A1C9 28 08         343          JR      Z,#LOAD4
A1CB CD 9D 1F      344          CALL    #FPRINT
A1CE CD EE 1F      345          CALL    #LTNL
A1D1 18 F1         346          JR      #LOAD3
A1D3               347  #LOAD4
A1D3 2A 06 A0      348          LD      HL,(#TEXTEND)
A1D6 E5            349          PUSH    HL
A1D7 2B            350          DEC     HL
A1D8 CD 4E A2      351          CALL    #UCHK1
A1DB E1            352          POP     HL
A1DC 20 01         353          JR      NZ,LOAD2
A1DE 2B            354          DEC     HL
A1DF               355  LOAD2
A1DF 22 70 1F      356          LD      (#DTADR),HL
A1E2 ED 5B 72 1F   357          LD      DE,(#SIZE)
A1E6 19            358          ADD     HL,DE
A1E7 38 20         359          JR      C,LOADERR2
A1E9 ED 5B 6A 1F   360          LD      DE,(#MEMAX)
A1ED B7            361          OR      A
A1EE ED 52         362          SBC     HL,DE
A1F0 30 17         363          JR      NC,LOADERR2
A1F2 11 68 A2      364          LD      DE,MMSTR3
A1F5 CD E5 1F      365          CALL    #MSX
A1F8 CD 9D 1F      366          CALL    #FPRNT
A1FB CD A6 1F      367          CALL    #RDD
A1FE CD EE 1F      368          CALL    #LTNL
A201 CD 5A A2      369          CALL    #ENDSEARCH
A204 D0            370          RET     NC
A205               371  LOADERR
A205 CD 33 20      372          CALL    #ERROR
A208 C9            373          RET
A209               374  LOADERR2
A209 11 79 A2      375          LD      DE,MMSTR5
A20C CD E5 1F      376          CALL    #MSX
A20F CD EE 1F      377          CALL    #LTNL
A212 C9            378          RET
A213               379
A213               380  ;TEXT SAVE
A213               381
A213               382  #SAVE
A213 13            383          INC     DE
A214 3E 04         384          LD      A,04
A216 CD A3 1F      385          CALL    #FILE
A219 21 00 00      386          LD      HL,0000
A21C 22 70 1F      387          LD      (#DTADR),HL
A21F 22 6E 1F      388          LD      (#EXADR),HL
A222 2A 06 A0      389          LD      HL,(#TEXTEND)
A225 ED 5B 04 A0   390          LD      DE,(#TEXTENT)
A229 D5            391          PUSH    DE
A22A B7            392          OR      A
A22B ED 52         393          SBC     HL,DE
A22D 23            394          INC     HL
A22E 22 72 1F      395          LD      (#SIZE),HL
A231 11 71 A2      396          LD      DE,MMSTR4
A234 CD E5 1F      397          CALL    #MSX
A237 CD 9D 1F      398          CALL    #FPRNT
A23A CD EE 1F      399          CALL    #LTNL
A23D CD AF 1F      400          CALL    #WOPEN
A240 38 08         401          JR      C,SERR
A242 E1            402          POP     HL
A243 22 70 1F      403          LD      (#DTADR),HL
A246 CD AC 1F      404          CALL    #WRD
A249 D0            405          RET     NC
A24A               406  SERR
A24A CD 33 20      407          CALL    #ERROR
A24D C9            408          RET
A24E               409
A24E               410  ;ADDRESS CHECK
A24E               411
A24E               412  #UCHK1
A24E ED 5B 04 A0   413          LD      DE,(#TEXTENT)
A252 B7            414          OR      A
A253 ED 52         415          SBC     HL,DE
A255 C8            416          RET     Z
A256 19            417          ADD     HL,DE
A257 F6 01         418          OR      1
A259 C9            419          RET
A25A               420
A25A               421  ;TEXT END ADDRESS CHECK
A25A               422
A25A               423  #ENDSEARCH
A25A 97            424          SUB     A
A25B 01 00 00      425          LD      BC,0000
A25E 2A 04 A0      426          LD      HL,(#TEXTENT)
A261 ED B1         427          CPIR
A263 2B            428          DEC     HL
A264 22 06 A0      429          LD      (#TEXTEND),HL
A267 C9            430          RET
A268               431
A268 4C 4F 41 44   432  MMSTR3  DB      "LOADING ",00
A26C 49 4E 47 20
A270 00
A271 53 41 56 49   433  MMSTR4  DB      "SAVING ",00
A275 4E 47 20 00
A279 4D 45 4D 4F   434  MMSTR5  DB      "MEMORY OVER",00
A27D 52 59 20 4F
A281 56 45 52 00
A285               435
```

投稿プログラム大募集
のお知らせ

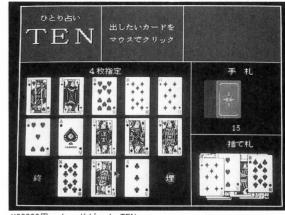
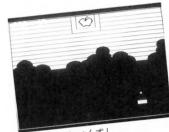
●あなたはどのようにしてプログラムに出会いましたか?

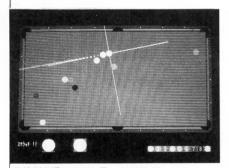
記号の羅列にすぎなかったプログラムリストが突然意味を持ったメッセージとして読み取れる,それを機に「プログラム」というものについてなにか納得できるようになる……。きっかけは雑誌のページの隅に載った小さな小さなプログラムだったのかもしれません。またはいくら見直してもエラーの出てくる長いBASICプログラムかもしれません。きっとそのプログラムにある「なにか」に魅かれてリストを打ち込んだことがあると思います。

あるソフトを使っていて,なにかの記事を読んでいて,または突然に,「こんなソフトがあったらいいな」と思う。こういった小さな動機からプログラムは生まれてきます。あなたのアイデアを埋もれさせないでください。私たちはそこにある「なにか」を求めています。完成度の高いありふれたプログラムよりも,粗削りでもオリジナリティの光るプログラムのほうが,さらに誰かの「プログラム」を生むことになるはずです。

Oh!Xには毎月さまざまな投稿プログラムが掲載されています。これらのプログラムは,すべて読者の皆さんが日頃のパーソナルコンピューティングのなかで作り上げてきたものです。あなたも投稿プログラムを通じてOh!Xの誌面作りに参加しませんか?

●大作歓迎!

Oh!Xでは過去に40Kバイト程度のプログラムまで誌上に掲載した実績があります。また,どうしても誌面に載り切らない作品は付録ディスクに収録して配布したこともありました。どうせ誌面には掲載できないからと諦めている方,とりあえずご連絡ください。

X68000用 カードゲーム TEN

X68000用 かべくずし

MZ-700用 Eyelarth

全機種共通システム用 BILLIARDS

X68000用 XROT0. X

X68000用 ハンディイメージスキャナアダプタの製作

投稿募集要項

1) お送りいただくプログラムには,住所,氏名,年齢,職業,連絡先電話番号,機種名,使用言語,動作に必要な周辺機器,マイコン歴などを明記のうえ,封書の宛先の最後には「Oh!X LIVE」,「全機種共通システム」,「投稿ゲームプログラム」など,プログラムの内容を明確にご記入ください。

2) 投稿されるプログラムには詳しい内容を記入した原稿と一緒に変数表,メモリマップ,参考文献などもお書き添えのうえお送りください。また,お送りいただいた原稿については,当方で加筆修正をさせていただくことがあります。

3) お送りいただくプログラムは最低2回はセーブしておいてください。基本的に同封されたフロッピーディスク,カセットテープ,クイックディスクなどについてはご返送いたしませんので,あらかじめご了承ください。

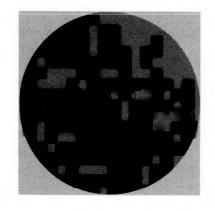
4) ハード製作関係の投稿につきましては,最初は内容のわかる原稿のみお送りいただければ結構です。その後,当方で製作物が必要だと判断した場合には改めて連絡いたします。

5) お送りいただいた作品の採用につきましては,掲載号が決定した時点で当方より連絡いたします。特にツール関係,ハード関係などのものにつきましては特集内容などを考慮したうえで採用決定されますので,結果を連絡するまでにかなり時間がかかる場合があります。

6) 投稿いただいたプログラムにバグなどが発見された場合は新しいプログラムの入ったメディアと一緒に文書にてご連絡ください。

7) 掲載されたプログラムに対しては当社規定の原稿料をお支払いいたします。また,投稿されたプログラムの著作権などは制作者に保留されますが,PDSなどとしてネットにアップロードされる場合は必ず事前に編集部までご連絡ください。なお,一般的モラルとして,他誌との二重投稿または他誌に掲載されたプログラムの移植などについては固くお断りいたします。

宛先

〒108 東京都港区高輪2-19-13 NS高輪ビル
ソフトバンク株式会社 Oh!X編集室「投稿プログラム」係

メニューによるファイル管理

亀田 雅彦 Kameda Masahiko

わかりやすいメニュー形式でファイル操作や実行を行うユーティリティです。いろいろと工夫されているので機能も豊富。これまでに作ったKAME-DOS用のコマンドなども起動できます。

今月はMENUだ

MENUといっても，Human Ver.2.0についてるようなのじゃなくて，「ファイルセレクタ」みたいなやつです。「COMMAND.X1」ではファイル名をキーボードから入力しますが，それを「カーソルで選択できるようにした」という代物です。

X68000のPDSでよく見かけていて，ちょいと作りたくなってしまいました。簡単にできるかなと思ってたけど，意外にてこずったかな。自分で作ったKAME-DOSながらもう中身を忘れているし，しかも，BASICのメモリ不足には泣かされるし。でも，それだけにかなりの力作だと思います。操作性はほぼ完璧でしょう。はっはっは（自我自賛モード）。

余談はこのくらいにして，今回はX1シリーズ全機種対応です。機能は同じですが，プログラムはノーマルX1用とturbo用に分かれているので，打ち込むときは注意してください。もちろん実行には，いままでの外部コマンドと同様，「INTEGRAL X一式」が必要です。

機能

まず，機能から説明しましょう。「MENU.X1」は，「COMMAND.X1」の代表的な機能の一部をより簡単に実行するために作られました。したがって「COMMAND.X1」に完全に置き換わるものではなく，補佐的役割をはたします。それに，ファイルが4つに分かれるため「機動性が悪い」という面もあります。詳しくはまたあとで。

●ファイルセレクト実行

「COMMAND.X1」には，外部コマンドおよび普通のBASICファイルを，コマンドラインから実行する機能がありました。そのファイル名入力部分を，カーソル選択で行います。また，パラメータとして渡すファイル名もカーソルでセレクトするので，キー入力する必要はありません。

●ドライブ/ディレクトリの移動

ドライブはポップアップ式メニューで，ディレクトリはファイル名と同じようにセレクトして移動します。画面には常にカレントディレクトリのファイル名一覧が表示されていて，スクロールさせて見ることができます。

●COPY, DEL

それぞれ対象ファイルをセレクトしておいてから，命令を実行します。選択方法はワイルドカード的な選択だけでなく，複数の任意のファイルをチョイスできるようにしたので，便利に使えると思います。

＊　＊　＊

その他，ディスクタイプ自動判別，コンバート機能などは，「COMMAND.X1」と同様にサポートしました。逆にサポートされていないのは，デバイスドライバ，リダイレクション，TYPE命令などです。

結局，価値はどのへんにあるのか？　というと，「片手でジュースを飲みながら，マウスならぬテンキーで操作ができること」でしょう。あと，ファイルの実行が簡単なので，いろんな外部コマンドが考えられるようになったことかな。

入力方法

リスト1：共通・BASIC
リスト2：共通・BASIC
リスト3：turbo用・マシン語
リスト4：turbo用・マシン語
リスト5：ノーマルX1用・マシン語
リスト6：ノーマルX1用・マシン語

turbo用というのはturboBASIC，Z-BASICで使用する場合。CZ8FB01 ver.1.0で使うときはturboでもX1用のリストを使います（以下同様）。

まず，リスト1，2はturbo，X1共通なので，それぞれのBASICで打ち込んでください。ただし，リスト2はX1turboの場合そのまま入力しますが，X1の場合は一部変更します。いままでの外部コマンドと同じように，リスト2の最後のDATA文が英語と日本語で対になっています。X1ならこの英語のDATA文の注釈（'）をはずして，日本語のDATA文の部分は入力しないようにします。その入力結果を図1に示しておきます。それからX1の場合，すべての注釈行のコメントは入力しないでください（注釈行自体は入力します）。

上記のようにリスト1，2を打ち込んだら，

リスト1：SAVE “ME.BAT”
リスト2：SAVE “MENU.X1”

というファイル名でセーブします。

マシン語リストはX1turbo用，X1用それぞれ2つずつあります。X1turboならリスト3，4，X1ならリスト5，6をそれぞれのマシン語入力ツールから入力します。チェックサムを確認したら，

CLEAR　&HC200 (turbo)

または，

CLEAR　&HF200 (X1)

を実行して，

リスト3：SAVEM “MENU.OBJ”, &HC200, &HCAD1
リスト4：SAVEM “KEY.OBJ”, &HCE80, &HCEFF

または，

リスト5：SAVEM “MENU.OBJ”, &HF200, &HFAD6
リスト6：SAVEM “KEY.OBJ”, &HFE80, &HFEFF

としてセーブしてください。

最終的にBASIC 2つ，マシン語2つの計4つのファイルができます。この4つのファイルは一緒に，同一のルートディレクトリにセーブするようにしてください。

なお，テストRUNするときには，

・必ずKAME-DOSの「COMMAND.X

1」を起動してから行う
・変数名ひとつ違うだけでも暴走したりファイルを壊したりする可能性がある
などの注意点は，前回までの外部コマンドと同じです。バックナンバーもよく読んでおいてください。

使い方

●起動方法

まずはKAME-DOSを立ち上げます。

「ME．BAT」
「MENU．X1」
「MENU．OBJ」
「KEY．OBJ」

の4つのファイルが揃っていることを確認したら，コマンドラインから，

ME［RET］

と打ち込んでください。順にファイルがロードされて，図2に説明されているような画面になります。「ME　A：」「ME　ファイル名」のように，パラメータやオプションをつけることはできません。また，「MENU．X1」には「X1」の拡張子がついていますが，最初の起動/再起動にかかわらず「ME．BAT」から立ち上げてください。

●ファイルセレクト

起動直後の状態では，反転カーソルがファイル名上にあります。もしこのときエラーがあればエラーメッセージを表示します。このメッセージはなにかキーを押せば消えますが，ドライブ変更などをしない限りエラー状態は変わりません。

テンキーの‘8’‘2’で，そのカーソルを上下に動かせます。ファイルが多くてWINDOWに入りきらないときは，8・2キーでスクロールします。また，‘7’‘1’キーを押すと，カーソルを上下に10ファイルずつ飛ばして移動できます。このファイルセレクトが，このプログラムでの最重要作業になります。

●ドライブセレクト

テンキーの‘4’‘6’を押してみてください。ファイル名WINDOWの上に，ポップアップ式にドライブセレクトメニューが表示されました。そのまま4，6を使って反転カーソルを左右に動かし，目的のドライブの上で4・6以外のキーを押してください。そのドライブのファイル名一覧が表示されます。

また，そのまま4，6を押し続けてメニューの外側に出せば，ドライブセレクトメニューはキャンセルされます。そのほかに，直接‘A’～‘F’‘W’～‘Y’を押してドライブを変更することもできます。

なおこのプログラムでは，A～F，W～Yの計9つのドライブしかサポートしていません。それぞれのドライブがどの物理デバイスなのかは，バックナンバーを見てください。

●ディレクトリ移動

ディレクトリ名は，ファイル名に混じって一緒に表示されます。そのディレクトリ名にカーソルを合わせて‘リターンキー’を押せば下位ディレクトリに移動します。また，下位ディレクトリから上のディレクトリに戻るには，ファイル名の先頭にある「..」というファイル上でリターンキーを押します。テンキーの‘.’を押すことで，直接戻る方法もあります。

●マークファイル

マークファイルとはファイル名を一時的に記憶しておいて，あとでCOPY，DEL，実行などの対象ファイルとして使うものです。「COMMAND.X1」ではユーザーが1つひとつキータイプしていたファイル名を，まとめて指定できる利点があります。

テンキーの‘5’を押すと，ファイル名がマゼンタになり，頭に「<MRK>」がつきます。これがそのファイルをマークした印で，同時に複数個指定することができます。

ドライブ，ディレクトリを変更しても保存されますが，変更先で新たにマークすると元のマークは取り消されます。つまり，同時に同一のディレクトリにしか指定できないということです。左下の画面最下行に，マークファイルがあるかどうかを示します。「M」と出ているときは「マークがどこかに残っている」，空白のときは「マークがない」ということです。

‘＋’キーで現在表示中の全ファイルをマーク。‘－’キーはその逆，全マークを取り消します。

●集約機能

‘/’（スラッシュ）キーには，3つの機能（COPY，DEL，ワイルドカード）が割り当ててあります。1回押せばCOPY，もう1回押すとDEL，3回でワイルドカードの実行選択メニューがポップアップします。それぞれのときにリターンキーを押せば実行，スラッシュ・リターン以外のキーでキャンセルです。

1）　COPY

COPYは以下の手順で行います。

・コピーしたいファイルをマークする（同一ディレクトリなら複数個可）
・コピー先となるドライブ&ディレクトリに移動する
・スラッシュを1回押して，COPYを実行する

図1　リスト2のX1用変更点最終入力結果

```
3310 '
3320 '
3330 DATA COPY <mark> to current-dir
3350 DATA DELETE <mark> files
3370 DATA INPUT FILE-NAME for print
3390 DATA Error !!
3410 DATA Drive not ready
3430 DATA FILE Not Found
3450 DATA No exec file
3470 DATA Not REBOOT
```

図3　キー操作一覧表

キー	機能
8，2	ファイルセレクトのカーソル上下移動
7，1	8，2の高速上下移動
4，6	ドライブセレクトのカーソル左右移動
5	カーソル上ファイルをマーク
+，－	全ファイルマーク，全マーク解除
A～F W～Y	ドライブ直接変換指定キー
”　.”	上位ディレクトリへ戻る
”　/”	COPY，DEL，ワイルドカード指定
RET	ファイル名の上で押せばそのファイルを実行
スペース	MENU，X1の終了
＊	WIDTH40と80の切り替え

図2　MENU．X1　画面構成

「マークしたあと移動して COPY する」手順を間違えないでください。

2) DEL

DELの実行も同様です。

・消去したいファイルをマークする

・スラッシュを2回押し,DELを実行する

3) ワイルドカード

ワイルドカードを選択すると，ファイル名1行入力モードになります。ここでファイル名（ワイルドカードを含むことができる）を入力してください。現ディレクトリ内でそれに該当するファイル一覧を表示します。これと`+'キーのマーク機能を併用して，COPYなりDELなりすれば，ワイルドカード機能を実現できます。

●ファイル実行

COMMAND.X1では，ファイル名をキータイプしてファイルを実行していました。MENU.X1からは，実行したいファイルにカーソルを合わせて‘リターンキー’を押すことで，そのファイルを実行します。また，パラメータとして渡すファイル名も，マークを利用して指定できます。方法は，

・「実行したいファイル」をマークする

・パラメータとして渡すファイルのあるディレクトリへ移動する

・そのファイル上でリターンキーを押す

例を挙げますと，「FORMAT.X1」「DISKCOPY.X1」などのようにファイル名を渡さない外部コマンドの場合，単純にそのファイル名の上でリターンキーを押します。

「GLOAD.X1」「XLOAD.X1」のようにファイル名を渡す場合（画像ファイルを「GAZO.GH1」とします)。まず「GLOAD.X1」をマークします。次に「GAZO.GH1」上でリターンキーを押します。これでコマンドラインから,

GLOAD　GAZO.GH1

を実行したのと同じことになります。そして一度この作業を行えば，2度目からは「GAZO.GH1」上でリターンキーを押すだけで済みます（マークする必要がない)。前に実行したときのマークファイルを記憶しているからです。この記憶はなんらかの外部コマンドが実行されるまで保持されます。また次に「XLOAD.X1」などの，他のファイル名を渡す外部コマンドを実行するときは，マークするところからやり直してください。

なお，「GSAVE.X1」「XSAVE.X1」のように新規のファイル名を渡す外部コマンドに対して，それを指定するような機能はありません。つまり，これらの外部コマンドはMENU.X1からは実行できません。

●その他

‘スペースキー’でMENU.X1を終了して，親プロセスへ戻ります。MENU.X1からMENU.X1を呼び出すような2重呼び出しはしないようにしてください。

‘＊'キーで「WIDTH 40 OR 80」の切り替えができます。例のごとく，どちらでも動くように設計されているので，見やすいほうで使ってください。「ME.BAT」には，起動時のWIDTHが設定されているので変更可です。画面の低/高解像度は，もともとのKAME-DOSで設定されているものに従います。

最初に書いたとおり，このプログラムは4つのファイルで構成されているので,「COMMAND.X1」さえあれば動くという状態より「重く」なっています。それにコマンドラインのようにきめ細かな対応もできません。たとえばコピーの際，オプションがつけられない，コピー先のファイル名が指定できないなどです。つまり，MENU.X1とCOMMAND.X1は，利用目的に応じてユーザーが使い分けることを前提としているのです。

＊

いろいろ書きましたが，感覚的にわかりやすい画面構成なので，とりあえず触ってみればすぐのみ込めます。大切なのは，前記の「マークファイル」「集約機能」「ファイル実行」ぐらいで，ほかは一読すれば十分でしょう。キー操作の一覧表も載せておくので，忘れてしまったときにでも使ってください。

今月は実用本位の設計思想なんですが，操作性や機能など満足していただけたでしょうか？　BASICの変数名にまで気を使って高速化しているので，小気味よいテンポは間違いないと思います。それから，今回のプログラム作成に当たって，X68000のいくつかのPDSを参考にさせていただきました。この場を借りてお礼申し上げます。

リスト1

```
1000 'MENU.X1  Ver 1.0       By Kameda
1010 '
1020 IF PEEK(&HD07F) THEN s=&HC200:CLEAR s:KLIST 0 ELSE s=&HF200
1030 '
1040 DEFUSR1=m_opens:DEFUSR2=m_preop:DEFUSR3=m_tranr
1050 '
1060 LOADM "MENU.OBJ":LOADM "KEY.OBJ"
1070 DEFUSR0=s+6:DEFUSR4=s+&HC:DEFUSR5=s+&H18
1080 mc=s      :cp=s+3   :fw=s+9   :un=s+&HF
1090 mk=s+&H12:ck=s+&H15:tw=s+&H1B:cl=s+&H1E
1100 am=s+&H21:ac=s+&H24:kn=s+&H27:dl=s+&H2A
1110 co=s+&H2D
1120 '
1130 i0=s+&H40:du=s+&H68:cr=s+&H69:cb=s+&H6A
1140 de=s+&H6B:ys=s+&H6C:uu=s+&H6D:xx=s+&H6E
1150 wi=s+&H6F:ww=s+&H70:us=s+&H75:yy=s+&H76
1160 nw=s+&H78:ss=s+&H7A:c2=s+&H7C
1170 '
1180 i1=s+&HC58:i2=s+&HC80:i3=s+&HB48:sfe$="":CALL cl
1190 '
1200 IF widts<80 THEN bxx=0 ELSE bxx=basx
1210 byy=0:POKE xx,bxx:POKE yy,byy
1220 POKE us,5:POKE ys,17:POKE uu,1:POKE wi,widts
1230 CONSOLE 0,24:GOSUB 3030
1240 POKE v_dn,PEEK(s_dn):POKE v_ddrv+1,7,1:KEY0,""
1250 '
1260 'MAIN
1270 fe$=""
1280 od=1:sb=0:op=1:GOSUB 2970
1290 es=0:k=PEEK(v_stop):t=PEEK(v_csdir+PEEK(v_dn))
1300 st=t:sd=PEEK(v_dn):sk=sd:IF sk>21 THEN sk=sk-15
1310 a40=i1+sk*4+t:MEM$(ss,4)=MKI$(a40)
1320 IF k=3  AND t=0  THEN es=1
1330 IF K<>0 AND k<>3 THEN es=2
1340 POKE uu,1:POKE du,1
1350 IF t>0 THEN POKE uu,0:POKE du,0 ELSE POKE a40+1,0,0,0
1360 POKE cr,PEEK(du):POKE cb,1:POKE c2,PEEK(du):GOSUB 2770
1370 IF es ELSE CALL cp
1380 REPEAT:d$=INKEY$:UNTIL d$<>"":KEY0,"":k=ASC(d$)
```

```
1390 ON PEEK(i2+k)-90 GOTO 1500,1530,1530,2330,2080,1670
1400 IF es GOTO 1380
1410 ON PEEK(i2+k) GOTO 1460,1470,1480,1490,1500,1570,1550,1620,1640,1650
1420 GOTO 1370
1430 GOTO 1280
1440 GOTO 1260
1450 'f1
1460 d$=USR0(CHR$(   1)):GOTO 1420
1470 d$=USR0(CHR$(&HFF)):GOTO 1420
1480 d$=USR0(CHR$(  10)):GOTO 1420
1490 d$=USR0(CHR$(&HF6)):GOTO 1420
1500 '
1510 IF k=>ASC("a") THEN k=k-ASC("a") ELSE k=k-ASC("A")
1520 POKE v_dn,k:POKE s_dn,k:GOTO 1440
1530 '
1540 KEY0,d$:GOSUB 2140:IF s OR es THEN 1440 ELSE 1420
1550 '
1560 IF st GOSUB 1960:GOTO 1440 ELSE 1420
1570 '
1580 CALL fw:IF PEEK(ww+3)=4 GOSUB 1910:GOTO 1440
1590 CALL kn:IF PEEK(v_yen)<>32 GOTO 1730
1600 IF PEEK(ww+3)=1 GOTO 1800
1610 IF fe$(0)="" GOSUB 3230:GOTO 1420 ELSE 1700
1620 '
1630 CALL mk:d$=USR0(CHR$(1)):CALL kn:GOTO 1420
1640 CALL am:GOTO 1660
1650 CALL ac
1660 CALL kn:s=i3+116:fe$=MEM$(s+1,PEEK(s)):GOTO 1430
1670 '
1680 widts=-(widts=>80)*40-(widts<80)*80:CLS:WIDTH widts:GOTO 1190
1690 '
1700 'ex
1710 POKE ww+2,1:d$=fe$(0)
1720 s=i3+16:fe$(1)=MEM$(s+1,PEEK(s))+MEM$(i0+7,17):GOTO 1850
1730 '
1740 s=i3+16:fe$(1)=MEM$(s+1,PEEK(s))+MEM$(i0+7,17)
1750 POKE cr,1:CALL tw:IF PEEK(v_stop) GOTO 1440
1760 IF PEEK(ww+3)<>1 GOSUB 3230:GOTO 1440
1770 k=PEEK(v_mac):t=PEEK(v_dn):GOSUB 2010
1780 IF PEEK(&HD07F) THEN s=i3+144:d$=d$+MEM$(s+3,PEEK(s)-2)
1790 fe$(0)=d$+MEM$(i0+7,17):GOTO 1840
1800 '
1810 fe$=MEM$(i0+7,17):GOSUB 2990:IF PEEK(v_stop) GOTO 1440
1820 k=PEEK(v_mac):t=sd:GOSUB 2010:fe$(0)=""
1830 IF PEEK(&HD07F) THEN d$=d$+y$+dir$(t)
1840 d$=d$+MEM$(i0+7,17)
1850 proces$(proces)="MENU.X1":proces=proces+1
1860 CONSOLE 0,24:LOCATE 0,23
1870 IF PEEK(&HD07F) THEN CLEAR &HD000
1880 IF PEEK(ww+2)=0 GOTO 1900
1890 CHAIN d$
1900 KEY0,CHR$(26)+"RUN"+CHR$(34)+d$+CHR$(13):END
1910 '
1920 fe$=MEM$(i0+7,13):IF PEEK(cr)=0 GOTO 1960
1930 POKE v_cdf,1:GOSUB 2990:POKE v_cdf,0
1940 d$=MEM$(v_p256+1,PEEK(v_p256))
1950 k=PEEK(v_dn):dir$(k)=dir$(k)+d$:RETURN
1960 '
1970 fe$="":k=PEEK(v_dn):s=LEN(dir$(k))
1980 w=1:REPEAT:t=w:w=INSTR(t+1,dir$(k),y$):UNTIL w=s
1990 IF t=1 THEN dir$(k)="" ELSE dir$(k)=LEFT$(dir$(k),t)
2000 POKE a40,0:POKE v_csdir+k,st-1:RETURN
2010 '
2020 d$=STR$(t)+":"
2030 IF t<4 THEN IF k=1 THEN DEVICE d$+"2" ELSE DEVICE d$+"0"
2040 IF t=4 THEN d$="MEM0:"
2050 IF t=5 THEN d$="MEM1:"
2060 IF t>21 THEN d$="EMM"+RIGHT$(STR$(t-22),1)+":"
2070 DEVICE d$:RETURN
2080 '
2090 d$=USR3(proces$(proces-1)):IF PEEK(v_stop) GOSUB 3240:GOTO 1440
2100 k=PEEK(v_dn):IF k<4 THEN DEVICE STR$(k)+":"+RIGHT$(STR$(3-PEEK(v_mac)),1)
2110 CONSOLE 0,24:LOCATE 0,23:proces=proces-1
2120 IF PEEK(&HD07F) THEN CLEAR &HD000 ELSE CLEAR
2130 CHAIN MEM$(v_p256+&H81,PEEK(v_p256+&H80))
2140 '
2150 d$=USR4(CHR$(1,7,38,4)):i=bxx+2:w=byy+9
2160 LOCATE i+12,w-1:COLOR 5:CREV 1:PRINT "*DRIVE SELECT*";
2170 COLOR 7:LOCATE i,w
2180 PRINT " A:  B:  C:  D:  E:  F:  W:  X:  Y:";:CREV 0
2190 s=PEEK(v_dn)+1:IF s>22 THEN s=s-16
2200 t=i+4*(s-1)
2210 LOCATE t,w:PRINT SCRN$(t,w,4);
2220 REPEAT:d$=INKEY$:UNTIL d$<>"":KEY 0,""
2230 IF d$<>"4" AND d$<>"6" GOTO 2290
2240 LOCATE t,w:CREV 1:PRINT SCRN$(t,w,4);:CREV 0
2250 IF d$="4" THEN t=t-4:s=s-1
2260 IF d$="6" THEN t=t+4:s=s+1
2270 IF s<1 OR s>9 GOTO 2310
2280 GOTO 2210
2290 IF s>6 THEN s=s+16
2300 CALL un:POKE v_dn,s-1:POKE s_dn,s-1:RETURN
2310 CALL un:s=0:RETURN
2320 '
2330 'ESC
2340 w=1:RESTORE 3320
2350 d$=USR4(CHR$(4,7,32,5)):s=bxx+4:t=byy+7
2360 READ d$:LOCATE s+2,t+1:CREV 1:COLOR 5
2370 PRINT LEFT$(d$+STRING$(28," "),28);:COLOR 7:CREV 0
2380 LOCATE s+11,t+3:CREV 1:COLOR 6:PRINT "Yes:[RET]":COLOR 7:CREV 0
2390 KEY0,"":REPEAT:d$=INKEY$:UNTIL d$<>""
2400 IF d$=CHR$(47) THEN w=w+1:IF w<4 GOTO 2360
2410 IF d$<>CHR$(13) THEN w=0:CALL un:GOTO 1420
2420 LOCATE s+2,t+3:PRINT STRING$(28," "):LOCATE s+2,t+3
```

▶11時30分発の電車に乗るとかわいい女の子が乗っているので，研究室に残ってしまう。
そして，X68000を動かす時間がなくなってしまう。　住田　弘祐(22)広島県

```
2430 ON w GOTO 2480,2460,2450
2440 '
2450 k=28:GOSUB 2710:GOTO 1430
2460 GOSUB 2510:IF PEEK(v_stop) GOSUB 3200
2470 GOTO 1440
2480 dn=PEEK(v_dn):GOSUB 2600:POKE v_dn,dn
2490 POKE v_ddrv+1,7,1:IF PEEK(v_stop) GOSUB 3200
2500 GOTO 1440
2510 'del
2520 x=POS(0):y=CSRLIN:POKE cr,1
2530 GOSUB 2680:IF PEEK(v_stop) OR PEEK(cr)>127 RETURN
2540 POKE cr,1
2550 LOCATE x+1,y:PRINT MEM$(i0+7,17);
2560 CALL dl:IF PEEK(v_stop) RETURN
2570 IF PEEK(cr)<=127 GOTO 2550
2580 POKE v_stop,0:CALL m_clos2
2590 RETURN
2600 'copy
2610 POKE cr,1:t=0:s=&H10:w=POS(0):i=CSRLIN
2620 '
2630 POKE v_ddrv+1,7,1
2640 GOSUB 2680:IF PEEK(v_stop) OR PEEK(cr)>127 RETURN
2650 LOCATE w+1,i:PRINT MEM$(i0+7,17);
2660 CALL co:IF PEEK(v_stop) RETURN
2670 GOTO 2620
2680 'SUB
2690 CALL tw:k=PEEK(v_dn):IF PEEK(v_csdir+k)=0 THEN dir$(k)=""
2700 RETURN
2710 '
2720 x=POS(0):y=CSRLIN:CONSOLE y,1,x,k
2730 KEY0,sfe$:fe$="":INPUT "",fe$
2740 CONSOLE 0,24
2750 sfe$=fe$:RETURN
2760 '
2770 'INIT
2780 CALL m_dirsb:POKE de,PEEK(v_yen)
2790 MEM$(nw,2)=MEM$(v_bf,2)
2800 d$=CHR$(65+sd)+":"+y$+dir$(sd)
2810 k=LEN(d$):MEM$(i3+16,k+1)=CHR$(k)+d$
2820 k=LEN(fe$):MEM$(i3+116,k+1)=CHR$(k)+fe$
2830 MEM$(i3+141,3)=CHR$(PEEK(v_yen))+MEM$(v_fcrs,2)
2840 w$=dms$(PEEK(v_mac)):s=INT((11-LEN(w$))/2)
2850 LOCATE bxx+1,byy+1:PRINT STRING$(11," ");
2860 LOCATE bxx+s+1,byy+1:PRINT w$;
2870 LOCATE bxx+13,byy+1:PRINT STRING$(26," ");
2880 LOCATE bxx+13,byy+1:PRINTUSING "####",PEEK(v_yen);
2890 COLOR 4:PRINT " FILES ";:COLOR 7
2900 PRINTUSING "#####",CVI(MEM$(v_fcrs,2));
2910 COLOR 4:PRINT " Clusters";:COLOR 7
2920 LOCATE bxx+2,byy+3:COLOR 6:PRINT LEFT$(d$+STRING$(37," "),37);:COLOR 7
2930 LINE (bxx,byy+PEEK(us))-(bxx+40,byy+PEEK(us)+PEEK(ys))," ",bf
2940 CALL kn:ON es GOTO 3220,3210
2950 d$=USR0(CHR$(PEEK(a40))):CALL mc:RETURN
2960 '
2970 'open
2980 d$=USR5(CHR$(1,od,sb,op)+fe$):GOTO 3000
2990 d$=USR5(CHR$(2,1)+fe$)
3000 k=PEEK(v_dn):IF PEEK(v_csdir+k)=0 THEN dir$(k)=""
3010 RETURN
3020 '
3030 '
3040 CLS:CGEN 1:d$=STRING$(11,"1"):w$=STRING$(26,"1")
3050 s=byy+PEEK(us)+PEEK(ys)+1
3060 LOCATE bxx,s:PRINT "0112 0";STRING$(27,"1");CHR$(33,34,35,36,37);"1";
3070 LOCATE bxx,byy  :PRINT "5";d$;"8";w$;
3080 LOCATE bxx,byy+1:PRINT "4";STRING$(11," ");"4";STRING$(26," ");
3090 LOCATE bxx,byy+2:PRINT ":";d$;"9";w$;
3100 LOCATE bxx,byy+3:PRINT "4";STRING$(38," ");
3110 LOCATE bxx,byy+4:PRINT "3";STRING$(38,"1");
3120 LINE (bxx+39,s)-(bxx+39,s),"2"
3130 LINE (bxx+39,byy  )-(bxx+39,byy  ),"6"
3140 LINE (bxx+39,byy+1)-(bxx+39,byy+3),"4"
3150 LINE (bxx+39,byy+2)-(bxx+39,byy+2),";"
3160 LINE (bxx+39,byy+4)-(bxx+39,byy+4),"7"
3170 CGEN 0:RETURN
3180 '
3190 '
3200 RESTORE 3380:w=15:GOTO 3250
3210 RESTORE 3400:w=8:GOTO 3250
3220 RESTORE 3420:w=8:GOTO 3250
3230 RESTORE 3440:w=8:GOTO 3250
3240 RESTORE 3460:w=8
3250 READ m$:s=LEN(m$):t=INT((40-s)/2)
3260 d$=USR4(CHR$(t-2,w,s+4,3))
3270 LOCATE bxx+t,byy+w+1:CREV 1:PRINT m$;:CREV 0
3280 KEY0,"":REPEAT:d$=INKEY$:UNTIL d$<>"":KEY0,d$
3290 CALL un:POKE v_stop,0:RETURN
3300 '
3310 '
3320 'DATA COPY <mark> to current-dir  'X1用 ' を取りturbo用の行は打ちこまない
3330 DATA マークファイルをコピーします 'turbo用 turboはそのまま打ちこむ
3340 'DATA DELETE <mark> files        '        ** 以下同じ **
3350 DATA マークファイルを消去します
3360 'DATA INPUT FILE-NAME for print
3370 DATA ファイル名を入力してください
3380 'DATA Error !!
3390 DATA エラーが発生しました！！
3400 'DATA Drive not ready
3410 DATA ドライブの準備ができていません
3420 'DATA FILE Not Found
3430 DATA ファイルがありません
3440 'DATA No exec file
3450 DATA 実行不可能なファイルです
3460 'DATA Not REBOOT
3470 DATA リブートできません
```

▶11月号の西川さんの記事の中には「©闇の血族：システムサコム」というのがたくさんありましたが，システムサコムからお金でも渡されたのですか？　木下　卓也(18)埼玉県

リスト2

```
1000 'ME.BAT  Ver 1.0       By Kameda
1010 '
1020 widts=80 '80 or 40
1030 basx=20  '0<=basx<=40 (only widts=80)
1040 '
1050 WIDTH widts:SCREEN:fe$(0)=""
1060 RESTORE 1080:REPEAT:READ i,d$:DEFCHR$(i)=HEXCHR$(d$):UNTIL i=59
1070 CHAIN "MENU.X1"
1080 '
1090 DATA 32,000000000000000000000000000000000000000000000000
1100 DATA 33,00BD665A5E5A66BD00BD0000000000000000BD665A5E5A6600
1110 DATA 34,00FF9DDBD7CBDCFF00FF0000000000000000FF9DDBD7CBDC00
1120 DATA 35,00FFF9F5EDDDB8FF00FF0000000000000000FFF9F5EDDDB800
1130 DATA 36,00FF78556D7D7DFF00FF0000000000000000FF78556D7D7D00
1140 DATA 37,00FF03BF83BF81FF00FF0000000000000000FF03BF83BF8100
1150 DATA 48,007F7F7F7F7F7F7F007F40404040404040007F7F7F7F7F7F40
1160 DATA 49,00FFFFFFFFFFFFFF00FF0000000000000000FFFFFFFFFFFF00
1170 DATA 50,00FFFFFFFFFFFFFF00FE0000000000000000FEFEFEFEFEFE00
1180 DATA 51,7F7F7F7F7F7F7F7F40414040404040407E7F7F7F7F7F7F40
1190 DATA 52,7F7F7F7F7F7F7F7F40404040404040407E7E7E7E7E7E7E7E
1200 DATA 53,007F7F7F7F7F7F7F007F40404040404040007F7F7F7F7F7F7E
1210 DATA 54,00FFFFFFFFFFFFFF00FE0000000000000000FEFEFEFEFEFE3E
1220 DATA 55,7FFFFFFFFFFFFFFF40C00000000000007EFEFEFEFEFEFE00
1230 DATA 56,00FFFFFFFFFFFFFF00FF0000000000000000FFFFFFFFFFFF3E
1240 DATA 57,7FFFFFFFFFFFFFFF40C00000000000007EFEFFFFFFFFFF00
1250 DATA 58,7F7F7F7F7F7F7F7F40414040404040407E7F7F7F7F7F7F7E
1260 DATA 59,7FFFFFFFFFFFFFFF40C00000000000007EFEFEFEFEFEFE3E
```

リスト3

```
C200 C3 DA C6 C3 00 C7 C3 3D : ED
C208 C7 C3 AD C7 C3 97 C5 C3 : E0
C210 58 C6 C3 6C C5 C3 06 C5 : A0
C218 C3 6D C4 C3 DA C3 C3 BE : D5
C220 C3 C3 35 C5 C3 95 C9 C3 : 64
C228 76 C3 C3 51 C3 C3 89 C2 : 1E
C230 94 20 50 41 54 48 20 65 : 66
C238 72 72 6F 72 00 11 00 B8 : 8E
C240 0B 94 20 42 61 64 20 52 : 38
C248 45 43 4F 52 44 00 16 00 : 83
C250 C2 0B 94 20 52 65 73 65 : 10
C258 72 76 65 64 20 66 75 74 : 20
C260 75 72 65 00 0F 00 CC 0B : 32
C268 00 00 00 00 14 01 00 50 : 65
C270 45 43 00 07 00 03 00 00 : 92
C278 00 00 00 00 00 00 00 27 : 27
-----------------------------------
SUM: 22 F5 7E A1 76 C8 AD D2 C540

C280 00 34 00 EA 0B 94 20 00 : DD
C288 00 AF 32 87 C2 3E 10 32 : AA
C290 88 C2 3A 97 E0 32 7D C2 : 6C
C298 3E 07 32 8B D1 32 8C D1 : 62
C2A0 2A 78 E0 22 7F C2 2A 7A : 89
C2A8 E0 22 81 C2 3E 03 32 83 : 3B
C2B0 C2 3E 02 32 84 C2 3E 01 : B9
C2B8 32 85 C2 3E 03 32 86 C2 : 34
C2C0 11 83 C2 06 04 CD 18 C2 : 07
C2C8 3A 8C E0 B7 C0 3A 97 E0 : CE
C2D0 32 7E C2 CD 0B C3 CD 22 : FC
C2D8 C3 3A 88 C2 32 E5 EC AF : F9
C2E0 32 9C E0 CD 0C D0 3A 8C : 1D
C2E8 E0 B7 C0 3A 9C E0 B7 20 : E4
C2F0 F2 CD 39 C3 3A 81 D0 32 : 78
C2F8 82 D0 3A 87 C2 B7 28 03 : B7
-----------------------------------
SUM: 8A C0 C2 84 67 86 AA D9 FC4A

C300 32 82 D0 3E 02 32 80 D1 : 47
C308 C3 15 D0 3A 7D C2 E6 01 : 08
C310 C0 3A 7E C2 E6 01 C8 3E : 27
C318 04 32 87 C2 3E 12 32 88 : 89
C320 C2 C9 3A 7D C2 E6 01 C8 : B3
C328 3A 7E C2 E6 01 C0 3E 20 : 7F
C330 32 87 C2 3E 13 32 88 C2 : 48
C338 C9 3A E5 EC B7 C0 3A 87 : 0C
C340 C2 FE 20 C8 2A 7F C2 22 : 35
C348 86 D1 2A 81 C2 22 88 D1 : 3F
C350 C9 CD 03 E0 CD 12 D0 CD : F5
C358 15 C2 3A 69 C2 FE 80 D0 : 8A
C360 11 54 E0 21 47 C2 01 0D : 7D
C368 00 ED B0 21 55 C2 01 03 : D9
C370 00 ED B0 C3 06 D0 3A 6E : DE
C378 C2 C6 04 5F 16 00 3A 76 : B1
-----------------------------------
SUM: A9 5D 13 7F 63 A4 71 4D 4942

C380 C2 21 75 C2 86 21 6C C2 : EF
C388 86 3C 6F 26 00 CD 63 CA : 51
C390 C5 06 08 21 48 CD 7E 23 : AA
C398 B7 20 06 10 F9 16 20 18 : 34
C3A0 02 16 4D 7A 32 84 D1 C1 : 27
C3A8 3E 20 80 47 3E 03 ED 79 : CC
C3B0 3E 18 80 47 AF ED 79 78 : AA
C3B8 D6 08 47 ED 51 C9 11 49 : 86
C3C0 CD 21 48 CD 01 37 01 AF : EB
C3C8 77 ED B0 C9 11 49 CD 21 : 25
C3D0 48 CD 01 0F 00 AF 77 ED : 38
C3D8 B0 C9 DD 21 80 EE 3E 01 : 24
C3E0. DD 77 00 DD 77 01 DD 77 : FD
C3E8 03 AF DD 77 02 11 84 EE : 8B
C3F0 21 D8 CD CD 63 C4 21 3C : 17
C3F8 CE CD 63 C4 CD 58 C4 3A : E5
-----------------------------------
SUM: 23 48 69 B9 72 59 7E 5B 8BB5

C400 8C E0 B7 C2 CC C3 CD 2A : 6B
C408 D0 3A 84 D1 32 6B C2 2A : E8
C410 91 E0 22 78 C2 CD 15 C2 : 71
C418 CD 48 C4 3A 69 C2 FE 80 : BC
C420 D0 DD 21 80 EE 3E 01 DD : 58
C428 77 00 DD 77 01 DD 77 02 : 22
C430 AF DD 77 03 11 84 EE 21 : AA
C438 D8 CD CD 63 C4 21 47 C2 : C3
C440 01 11 00 ED B0 C3 58 C4 : 8E
C448 3A 84 D1 21 55 CE BE C8 : 59
C450 3E 80 32 69 C2 C3 CC C3 : 6D
C458 EB 11 80 EE B7 ED 52 45 : A5
C460 C3 18 C2 7E B7 C8 4F 06 : EF
C468 00 23 ED B0 C9 1A 13 05 : BB
C470 FE 02 28 25 FE 03 28 2A : A0
C478 CD C2 C4 CD D3 C4 C5 D5 : 51
-----------------------------------
SUM: 7A EE 81 27 BC 67 D2 F6 3A57

C480 CD 21 D0 D1 C1 3A 8C E0 : F6
C488 B7 C0 68 3A 84 D1 47 7D : 32
C490 90 6F 26 00 19 EB C3 24 : 10
C498 D0 1A 13 05 32 80 D1 C3 : 48
C4A0 21 D0 CD C2 C4 CD E6 C4 : BB
C4A8 CD D3 C4 3A 58 CD 47 11 : 1B
C4B0 59 CD CD 21 D0 CD F6 C4 : 6B
C4B8 3A 8C E0 B7 C0 06 00 C3 : E6
C4C0 24 D0 C5 3A 8B E0 4F 06 : B3
C4C8 00 21 C0 EC 09 7E 32 97 : 1D
C4D0 E0 C1 C9 1A 32 80 D1 13 : 1A
C4D8 05 1A 32 65 E0 13 05 1A : C8
C4E0 32 81 D1 13 05 C9 C5 D5 : FF
C4E8 21 68 E0 11 DD EE 01 23 : 69
C4F0 00 ED B0 D1 C1 C9 C5 D5 : 92
C4F8 11 68 E0 21 DD EE 01 23 : 69
-----------------------------------
SUM: D2 70 70 9F 62 42 6D 5A F5CD

C500 00 ED B0 D1 C1 C9 11 69 : 72
C508 C2 1A FE 01 20 09 32 6A : A0
C510 C2 2A 78 C2 22 C0 CA CD : 9F
C518 A3 C9 A6 20 0B 1A 21 55 : CD
C520 CE BE 30 0C 3C 12 18 E1 : 0F
C528 2F A6 77 CD 09 C2 B7 C9 : 64
C530 3E 80 12 37 C9 CD 7B C9 : E1
C538 30 0E 11 D8 CD 21 58 CD : 3A
C540 01 80 00 ED B0 CD 95 C9 : 49
C548 3E 01 32 69 C2 3A 55 CE : F9
C550 47 C5 CD 09 C2 3A 73 C2 : 13
C558 FE 04 28 08 11 69 C2 CD : 3B
C560 A3 C9 B6 77 21 69 C2 34 : 19
C568 C1 10 E6 C9 3A 73 C2 FE : ED
C570 04 C8 CD 7B C9 30 0E 11 : 2C
C578 D8 CD 21 58 CD 01 80 00 : 6C
-----------------------------------
SUM: 56 A4 47 16 1F 25 01 9E 75DD

C580 ED B0 CD 95 C9 11 69 C2 : 04
C588 CD A3 C9 57 A6 20 04 7A : D4
C590 B6 77 C9 2F A6 77 C9 D5 : E0
C598 DD E1 3A 6E C2 DD 86 00 : 8B
C5A0 5F 16 00 3A 76 C2 DD 86 : 4A
C5A8 01 6F 26 00 CD 63 CA ED : 7D
C5B0 43 B7 CA DD 7E 02 32 B9 : 0C
C5B8 CA DD 7E 03 32 BA CA ED : CB
C5C0 5B DF EC AF 21 00 38 CD : FB
C5C8 DB C5 3E 20 21 20 30 CD : 3C
C5D0 DB C5 3E 07 21 0F 20 CD : 02
C5D8 DB C5 C9 C5 32 BD CA 78 : 5F
C5E0 84 47 32 BC CA CD 2E C6 : 44
C5E8 CD ED C5 C1 C9 03 DD 6E : 57
C5F0 02 ED 78 CD 18 E0 13 3A : 79
C5F8 BD CA ED 79 03 2D 20 F1 : 2E
-----------------------------------
SUM: B6 DD 94 01 0D 2F EF 68 5FD1

C600 ED 4B B7 CA 3A BC CA 47 : C0
C608 DD 7E 02 6F 26 00 09 44 : 3F
C610 4D DD 6E 03 E5 3A 6F C2 : EB
C618 6F 26 00 09 44 4D E1 2D : 3D
C620 C8 ED 78 CD 18 E0 13 3A : 3F
C628 BD CA ED 79 18 E6 7D 32 : 9A
C630 BB CA DD 6E 03 DD 66 02 : 18
C638 C5 ED 78 CD 18 E0 13 3A : 3C
C640 BB CA ED 79 03 25 20 F1 : 24
C648 C1 E5 3A 6F C2 6F 26 00 : A6
C650 09 44 4D E1 2D 20 DE C9 : 6F
C658 ED 4B B7 CA ED 5B DF EC : CC
C660 3E 38 CD 70 C6 3E 30 CD : B4
C668 70 C6 3E 20 CD 70 C6 C9 : 60
C670 C5 80 47 32 BC CA CD B7 : C8
C678 C6 CD 7E C6 C1 C9 03 3A : 9E
-----------------------------------
SUM: 36 C3 DC E1 C3 16 F5 4F 9C78

C680 B9 CA 6F CD 15 E0 13 ED : B4
C688 79 03 2D 20 F6 ED 4B B7 : AE
C690 CA 3A BC CA 47 3A B9 CA : 8E
C698 6F 26 00 09 44 4D 3A BA : 23
C6A0 CA 6F E5 3A 6F C2 6F 26 : 1E
C6A8 00 09 44 4D E1 2D C8 CD : 3D
C6B0 15 E0 13 ED 79 18 EB 3A : AB
C6B8 BA CA 6F 3A B9 CA 67 C5 : DC
C6C0 CD 15 E0 13 ED 79 03 25 : 63
C6C8 20 F6 C1 E5 3A 6F C2 6F : 96
C6D0 26 00 09 44 4D E1 2D 20 : EE
C6D8 E2 C9 3A 68 C2 4F 3A 6C : 04
C6E0 C2 81 4F 3A 6B C2 B9 30 : E2
C6E8 01 4F 3A 68 C2 47 11 70 : 7C
C6F0 C2 79 B8 D8 78 12 13 AF : 17
C6F8 12 1B CD A6 C7 04 18 F1 : 74
-----------------------------------
SUM: 90 87 F5 32 BA 5C FB 7A 0E35

C700 D5 3A 68 C2 5F 3A 7C C2 : 10
C708 BB 38 14 57 3A 6C C2 83 : 49
C710 BA 38 0C 7A 11 70 C2 12 : CD
C718 13 AF 12 1B CD A6 C7 3A : 63
C720 69 C2 11 70 C2 12 13 3E : D1
C728 08 12 1B CD A6 C7 3A 6D : 16
C730 C2 57 3A 69 C2 92 ED 5B : 58
C738 7A C2 12 D1 C9 C5 D5 1A : 9C
C740 FE 80 D2 78 C7 4F 3A 69 : 81
C748 C2 81 32 69 C2 4F 3A 6B : 94
C750 C2 B9 30 03 32 69 C2 3A : 45
C758 69 C2 4F 3A 68 C2 47 3A : 5F
C760 6C C2 80 B9 D2 A3 C7 3A : DD
C768 6C C2 47 3A 69 C2 90 32 : 9C
C770 68 C2 CD 00 C2 C3 A3 C7 : E6
C778 3A 6D C2 47 3A 69 C2 4F : 64
-----------------------------------
SUM: 6F 75 EB 7D C4 46 0F 7B 1128

C780 1A 81 FE 80 30 08 B8 38 : 41
C788 05 32 69 C2 18 04 78 32 : 28
C790 69 C2 3A 68 C2 4F 3A 69 : 81
C798 C2 B9 D2 A3 C7 32 68 C2 : 13
C7A0 CD 00 C2 D1 C1 C9 CD E1 : 98
C7A8 C8 CD C6 C7 C9 D5 11 70 : 41
C7B0 C2 3A 69 C2 12 CD E1 C8 : AF
C7B8 3A 69 C2 B7 28 06 11 54 : AF
C7C0 C2 3E 2E 12 D1 C9 C5 D5 : 74
C7C8 E5 3A 68 C2 4F 1A 91 6F : B2
```

▶パソコン通信やりまくった。電話代メチャクチャ上がった。親は明細書見て泣きそうになった。次の日，キーボードが隠されていた。オレ，泣きそうや！　この状況バリバリ緊張した。
田中　哲(17)滋賀県

```
C7D0 3A 75 C2 85 6F 3A 76 C2 : D7
C7D8 85 6F 26 00 13 1A 32 71 : EA
C7E0 C2 3A 6E C2 5F 16 00 CD : 6E
C7E8 63 CA CD 84 CA DD 21 B2 : F8
C7F0 CA 3E 28 DD 77 00 CD 10 : 61
C7F8 C8 21 40 C2 DD 70 02 CD : 07
---------------------------------
SUM: F8 5D 47 9C B4 98 90 D5 564E

C800 55 C8 DD 46 02 23 03 DD : 45
C808 35 00 20 F0 E1 D1 C1 C9 : 81
C810 AF DD 77 03 DD 77 04 3A : 98
C818 71 C2 F6 07 DD 77 01 E6 : 6B
C820 08 C0 3A 73 C2 FE 04 3E : 77
C828 06 CA 51 C8 3A 74 C2 B7 : 10
C830 3E 03 C2 51 C8 3A 72 C2 : 8A
C838 B7 28 05 16 04 DD 72 03 : 50
C840 3A 73 C2 B7 C8 1E 05 FE : 0F
C848 03 28 02 1E 04 DD 73 04 : A3
C850 C9 DD 77 01 C9 3A B1 CA : 9C
C858 B7 28 10 F3 3E 1D D3 00 : 10
C860 7E CD 99 30 30 29 3E 1E : C9
C868 D3 00 FB DD 7E 00 DD 5E : 64
C870 01 DD 56 03 FE 13 CC D2 : E6
C878 C8 FE 12 CC D2 C8 FE 11 : 4D
---------------------------------
SUM: 84 64 03 87 B6 C1 54 AB 7C6D

C880 CC D2 C8 FE 08 CC D8 C8 : D8
C888 7B 5E 16 00 C3 B5 C8 56 : 85
C890 23 5E 2B CD 81 2F CD B6 : AC
C898 2F DD B6 01 CD B5 C8 DD : EA
C8A0 46 02 03 DD 70 02 23 DD : 9A
C8A8 35 00 CB F2 CD B5 C8 3E : 7A
C8B0 1E D3 00 FB C9 F5 F5 DD : 7C
C8B8 7E 02 C6 20 47 F1 ED 79 : 04
C8C0 DD 7E 02 C6 38 47 ED 51 : E0
C8C8 DD 7E 02 C6 30 47 ED 59 : E0
C8D0 F1 C9 7A B7 C8 5A AF C9 : 85
C8D8 DD 56 04 7A B7 C8 5A AF : 39
C8E0 C9 C5 D5 E5 ED 53 BE CA : 10
C8E8 21 40 C2 11 41 C2 01 27 : 5F
C8F0 00 3E 20 77 ED B0 ED 5B : BA
C8F8 BE CA 1A B7 20 06 CD C0 : 0C
---------------------------------
SUM: E0 6A A6 97 88 7D 5E 50 016E
```

```
C900 C9 C3 3D C9 FE 01 20 09 : BA
C908 2A 78 C2 22 C0 CA C3 2F : 02
C910 C9 5F 2A C0 CA 3A 6A C2 : 42
C918 FE 01 20 06 2A 78 C2 22 : AB
C920 C0 CA 93 28 0A 57 DC CF : 51
C928 C9 D4 E5 C9 22 C0 CA 11 : 08
C930 47 C2 CD FE C9 ED 5B BE : A3
C938 CA 1A 32 6A C2 CD 4F C9 : 27
C940 CD 09 CA ED 5B BE CA 1A : 8A
C948 32 7C C2 E1 D1 C1 C9 AF : 5B
C950 32 74 C2 3A 60 C2 FE 44 : 06
C958 21 C8 CA 28 15 CD 7B C9 : 01
C960 D8 ED 5B BE CA CD A3 C9 : E1
C968 A6 C8 21 CD CA 3E 01 32 : 97
C970 74 C2 11 41 C2 01 05 00 : 50
C978 ED B0 C9 21 58 CD 11 D8 : 95
---------------------------------
SUM: 85 FD 2E 27 B8 35 25 2C 6F68

C980 CD 06 80 1A BE 20 06 13 : 64
C988 23 10 F8 B7 C9 78 FE 04 : 25
C990 DC 95 C9 37 C9 11 49 CD : 61
C998 21 48 CD 01 0F 00 AF 77 : 6C
C9A0 ED B0 C9 1A CB 3F CB 3F : 94
C9A8 CB 3F 4F 06 00 21 48 CD : 95
C9B0 09 1A E6 07 0E 80 B7 28 : 7D
C9B8 05 CB 09 3D 18 F8 79 C9 : 68
C9C0 11 47 C2 3E 2E 12 13 12 : BD
C9C8 11 60 C2 3E 44 12 C9 F5 : 85
C9D0 7A ED 44 57 01 17 00 09 : 23
C9D8 CD 12 E0 23 FE 0D 20 F8 : 05
C9E0 15 20 F4 F1 C9 F5 01 19 : F2
C9E8 00 B7 ED 42 CD F5 C9 23 : 94
C9F0 15 20 F6 F1 C9 CD 12 E0 : A4
C9F8 FE 0D C8 2B 18 F7 CD 12 : EC
---------------------------------
SUM: 44 71 5C B2 38 77 E4 8E 937A

CA00 E0 FE 0D C8 23 12 13 18 : 13
CA08 F5 AF 32 72 C2 32 73 C2 : 71
CA10 3A 60 C2 FE 44 20 06 3E : 02
CA18 04 32 73 C2 C9 21 C2 CA : E1
CA20 CD 42 CA 21 C5 CA CD 42 : 98
CA28 CA 16 03 3A 60 C2 FE 4D : 8A
CA30 28 0B 16 01 FE 42 28 05 : B7
CA38 16 02 FE 41 C0 7A 32 73 : 36
```

```
CA40 C2 C9 C5 11 55 C2 06 03 : 81
CA48 1A 13 CD 5A CA BE 23 20 : 1F
CA50 07 10 F5 3E 01 32 72 C2 : B1
CA58 C1 C9 FE 61 D8 FE 7B D0 : 0A
CA60 D6 20 C9 3A 6F C2 FE 28 : 50
CA68 CA 78 CA 29 29 29 E5 : 95
CA70 29 29 19 D1 19 44 4D C9 : AF
CA78 29 29 29 E5 29 29 19 D1 : 9C
---------------------------------
SUM: 7E 43 AF BA A7 D5 16 45 D3C6

CA80 19 44 4D C9 C5 3A 77 C2 : AB
CA88 B7 3E 00 20 1F 3A 7F D0 : BD
CA90 B7 28 19 F3 3E 1D ED 79 : AC
CA98 3A 00 00 4F 3A 00 10 47 : 1A
CAA0 3E 1E D3 00 FB 78 B9 3E : 99
CAA8 01 20 01 AF 32 B1 CA C1 : 3F
CAB0 C9 00 44 4E 8B E0 02 06 : CE
CAB8 56 5F 53 54 4F 50 00 00 : FB
CAC0 00 00 58 31 20 42 41 54 : 80
CAC8 3C 44 49 52 3E 3C 4D 52 : 34
CAD0 4B 3E                   : 89
---------------------------------
SUM: A6 C9 72 FF C1 68 06 FD 2BF2
```

リスト4

```
CE80 00 00 00 00 00 00 00 00 : 00
CE88 00 00 00 00 00 06 00 00 : 06
CE90 00 00 00 00 00 00 00 00 : 00
CE98 00 00 00 00 00 00 04 03 : 07
CEA0 5F 00 00 00 00 00 00 00 : 5F
CEA8 00 00 60 09 00 0A 07 5E : D8
CEB0 00 03 01 00 5C 08 5D 04 : C9
CEB8 02 00 00 00 00 00 00 00 : 02
CEC0 00 5B 5B 5B 5B 5B 5B 00 : 22
CEC8 00 00 00 00 00 00 00 00 : 00
CED0 00 00 00 00 00 00 00 5B : 5B
CED8 5B 5B 00 00 00 00 00 00 : B6
CEE0 00 5B 5B 5B 5B 5B 5B 00 : 22
CEE8 00 00 00 00 00 00 00 00 : 00
CEF0 00 00 00 00 00 00 00 5B : 5B
CEF8 5B 5B 00 00 00 00 00 00 : B6
---------------------------------
SUM: 17 6F 17 BF 12 CE 1E 1B 2888
```

リスト5

```
F200 C3 D4 F6 C3 FA F6 C3 37 : 3A
F208 F7 C3 A7 F7 C3 97 F5 C3 : 6A
F210 55 F6 C3 6C F5 C3 06 F5 : 2D
F218 C3 6D F4 C3 DA F3 C3 BE : 35
F220 F3 C3 35 F5 C3 9A F9 C3 : F9
F228 76 F3 C3 51 F3 C3 89 F2 : AE
F230 00 00 00 00 00 00 00 00 : 00
F238 00 00 00 00 00 00 00 00 : 00
F240 00 00 00 00 00 00 00 00 : 00
F248 00 00 00 00 00 00 00 00 : 00
F250 00 00 00 00 00 00 00 00 : 00
F258 00 00 00 00 00 00 00 00 : 00
F260 00 00 00 00 00 00 00 00 : 00
F268 00 00 00 00 14 01 00 50 : 65
F270 00 00 00 00 00 03 00 00 : 03
F278 00 00 00 00 00 00 00 00 : 00
---------------------------------
SUM: 3B B0 4C 2F 56 A4 03 B2 AA9B

F280 00 00 00 00 00 00 00 00 : 00
F288 00 AF 32 87 F2 3E 10 32 : DA
F290 88 F2 3A 97 E0 32 7D F2 : CC
F298 3E 07 32 8B D1 32 8C D1 : 62
F2A0 2A 78 E0 22 7F F2 2A 7A : B9
F2A8 E0 22 81 F2 3E 03 32 83 : 6B
F2B0 F2 3E 02 32 84 F2 3E 01 : 19
F2B8 32 85 F2 3E 03 32 86 F2 : 94
F2C0 11 83 F2 06 04 CD 18 F2 : 67
F2C8 3A 8C E0 B7 C0 3A 97 E0 : CE
F2D0 32 7E F2 CD 0B F3 CD 22 : 5C
F2D8 F3 3A 88 F2 32 E5 EC AF : 59
F2E0 32 9C E0 CD 0C D0 3A 8C : 1D
F2E8 E0 B7 C0 3A 9C E0 B7 20 : E4
F2F0 F2 CD 39 F3 3A 81 D0 32 : A8
F2F8 82 D0 3A 87 F2 B7 28 03 : E7
---------------------------------
SUM: EA BC 52 2A BC 82 8A 69 E85C

F300 32 82 D0 3E 02 32 80 D1 : 47
F308 C3 15 D0 3A 7D F2 E6 01 : 38
F310 C0 3A 7E F2 E6 01 C8 3E : 57
F318 04 32 87 F2 3E 12 32 88 : B9
F320 F2 C9 3A 7D F2 E6 01 C8 : 13
F328 3A 7E F2 E6 01 C0 3E 20 : AF
F330 32 87 F2 3E 13 32 88 F2 : A8
F338 C9 3A E5 EC B7 C0 3A 87 : 0C
F340 F2 FE 20 C8 2A 7F F2 22 : 95
F348 86 D1 2A 81 F2 22 88 D1 : 6F
F350 C9 CD 03 E0 CD 12 D0 CD : F5
F358 15 F2 3A 69 F2 FE 80 D0 : EA
F360 11 54 E0 21 47 F2 01 0D : AD
F368 00 ED B0 21 55 F2 01 03 : 09
F370 00 ED B0 C3 06 D0 3A 6E : DE
F378 F2 C6 04 5F 16 00 3A 76 : E1
---------------------------------
SUM: 39 8D 73 DF F3 34 A1 7D E4A1

F380 F2 21 75 F2 86 21 6C F2 : 7F
F388 86 3C 6F 26 00 CD 68 FA : 86
F390 C5 06 08 21 48 FD 7E 23 : DA
F398 B7 20 06 10 F9 16 20 18 : 34
```

```
F3A0 02 16 4D 7A 32 84 D1 C1 : 27
F3A8 3E 20 80 47 3E 03 ED 79 : CC
F3B0 3E 18 80 47 AF ED 79 78 : AA
F3B8 D6 08 47 ED 51 C9 11 49 : 86
F3C0 FD 21 48 FD 01 37 01 AF : 4B
F3C8 77 ED B0 C9 11 49 FD 21 : 55
F3D0 48 FD 01 0F 00 AF 77 ED : 68
F3D8 B0 C9 DD 21 80 EE 3E 01 : 24
F3E0 DD 77 00 DD 77 01 DD 77 : FD
F3E8 03 AF DD 77 02 11 84 EE : 8B
F3F0 21 D8 FD CD 63 F4 21 3C : 77
F3F8 FE CD 63 F4 CD 58 F4 3A : 75
---------------------------------
SUM: B3 78 99 49 72 B9 E3 BB 9261

F400 8C E0 B7 C2 CC F3 CD 2A : 9B
F408 D0 3A 84 D1 32 6B F2 2A : 18
F410 91 E0 22 78 F2 CD 15 F2 : D1
F418 CD 48 F4 3A 69 F2 FE 80 : 1C
F420 D0 DD 21 80 EE 3E 01 DD : 58
F428 77 00 DD 77 01 DD 77 02 : 22
F430 AF DD 77 03 11 84 EE 21 : AA
F438 D8 FD CD 63 F4 21 47 F2 : 53
F440 01 11 00 ED B0 C3 58 F4 : BE
F448 3A 84 D1 21 55 FE BE C8 : 89
F450 3E 80 32 69 F2 C3 CC F3 : CD
F458 EB 11 80 EE B7 ED 52 45 : A5
F460 C3 18 F2 7E B7 C8 4F 06 : 1F
F468 00 23 ED B0 C9 1A 13 05 : BB
F470 FE 02 28 25 FE 03 28 2A : A0
F478 CD C2 F4 CD D3 F4 C5 D5 : B1
---------------------------------
SUM: 7A 1E 11 27 4C 27 02 B6 8AC6

F480 CD 21 D0 D1 C1 3A 8C E0 : F6
F488 B7 C0 68 3A 84 D1 47 7D : 32
F490 90 6F 26 00 19 EB C3 24 : 10
F498 D0 1A 13 05 32 80 D1 C3 : 48
F4A0 21 D0 CD C2 F4 CD E6 F4 : 1B
F4A8 CD D3 F4 3A 58 FD 47 11 : 7B
F4B0 59 FD CD 21 D0 CD F6 F4 : CB
F4B8 3A 8C E0 B7 C0 06 00 C3 : E6
F4C0 24 D0 C5 3A 8B E0 4F 06 : B3
F4C8 00 21 C0 EC 09 7E 32 97 : 1D
F4D0 E0 C1 C9 1A 32 80 D1 13 : 1A
F4D8 05 1A 32 65 E0 13 05 1A : C8
F4E0 32 81 D1 13 05 C9 C5 D5 : FF
F4E8 21 68 E0 11 DD EE 01 23 : 69
F4F0 00 ED B0 D1 C1 C9 C5 D5 : 92
F4F8 11 68 E0 21 DD EE 01 23 : 69
---------------------------------
SUM: D2 A0 A0 9F 92 72 6D BA 77AE

F500 00 ED B0 D1 C1 C9 11 69 : 72
F508 F2 1A FE 01 20 09 32 6A : D0
F510 F2 2A 78 F2 22 C5 FA CD : 34
F518 A8 F9 A6 20 0B 1A 21 55 : 02
F520 FE BE 30 0C 3C 12 18 E1 : 3F
F528 2F A6 77 CD 09 F2 B7 C9 : 94
F530 3E 80 12 37 C9 CD 80 F9 : 16
F538 30 0E 11 D8 FD 21 58 FD : 9A
```

```
F540 01 80 00 ED B0 CD 9A F9 : 7E
F548 3E 01 32 69 F2 3A 55 FE : 59
F550 47 C5 CD 09 F2 3A 73 F2 : 73
F558 FE 04 28 08 11 69 F2 CD : 6B
F560 A8 F9 B6 77 21 69 F2 34 : 7E
F568 C1 10 E6 C9 3A 73 F2 FE : 1D
F570 04 C8 CD 80 F9 30 0E 11 : 61
F578 D8 FD 21 58 FD 01 80 00 : CC
---------------------------------
SUM: F0 34 47 4B 0F 5A CB 8E C8CC

F580 ED B0 CD 9A F9 11 69 F2 : 69
F588 CD A8 F9 57 A6 20 04 7A : 09
F590 B6 77 C9 2F A6 77 C9 D5 : E0
F598 DD E1 3A 6E F2 DD 86 00 : BB
F5A0 5F 16 00 3A 76 F2 DD 86 : 7A
F5A8 01 6F 26 00 CD 68 FA ED : B2
F5B0 43 BC FA DD 7E 02 32 BE : 46
F5B8 FA DD 7E 03 32 BF FA ED : 30
F5C0 5B DF EC AF 21 00 38 3E : 6C
F5C8 20 21 20 30 CD D8 F5 3E : 69
F5D0 07 21 0F 20 CD D8 F5 C9 : BA
F5D8 C5 32 C2 FA 78 84 47 32 : 28
F5E0 C1 FA CD 2B F6 CD EA F5 : 55
F5E8 C1 C9 03 DD 6E 02 ED 78 : 3F
F5F0 CD 18 E0 13 3A C2 FA ED : BB
F5F8 79 03 2D 20 F1 ED 4B BC : AE
---------------------------------
SUM: F9 FF 21 DC EC 52 44 EC E028

F600 FA 3A C1 FA 47 DD 7E 02 : 93
F608 6F 26 00 09 44 4D DD 6E : 7A
F610 03 E5 3A 6F F2 6F 26 00 : 18
F618 09 44 4D E1 2D C8 ED 78 : D5
F620 CD 18 E0 13 3A C2 FA ED : BB
F628 79 18 E6 7D 32 C0 FA DD : BD
F630 6E 03 DD 66 02 C5 ED 78 : E0
F638 CD 18 E0 13 3A C0 FA ED : B9
F640 79 03 25 20 F1 C1 E5 3A : 92
F648 6F F2 6F 26 00 09 44 4D : 90
F650 E1 2D 20 DE C9 ED 4B BC : C9
F658 FA ED 5B DF EC 3E 38 3E : C1
F660 30 CD 6A F6 3E 20 CD 6A : F2
F668 F6 C9 C5 80 47 32 C1 FA : 38
F670 CD B1 F6 CD 78 F6 C1 C9 : 39
F678 03 3A BE FA 6F CD 15 E0 : 26
---------------------------------
SUM: AF 64 BD 9C 64 72 59 A5 CB74

F680 13 ED 79 03 2D 20 F6 ED : AC
F688 4B BC FA 3A C1 FA 47 3A : 77
F690 BE FA 6F 26 00 09 44 4D : E7
F698 3A BF FA 6F E5 3A 6F F2 : E2
F6A0 6F 26 00 09 44 4D E1 2D : 3D
F6A8 C8 CD 15 E0 13 ED 79 18 : 1B
F6B0 EB 3A BF FA 3A BE FA : 3F
F6B8 67 C5 CD 15 E0 13 ED 79 : 67
F6C0 03 25 20 F6 C1 E5 3A 6F : 8D
F6C8 F2 6F 26 00 09 44 4D E1 : 02
F6D0 2D 20 E2 C9 3A 68 F2 4F : DB
F6D8 3A 6C F2 81 4F 3A 6B F2 : FF
```

▶シムシティーの調査で線路を調べると人が住んでいた。公園にも住んでいるし……，まったく，お役所はおにしてんでしょう。　池田　孝志(16)奈良県

```
F6E0 B9 30 01 4F 3A 68 F2 47 : 14
F6E8 11 70 F2 79 B8 D8 78 12 : 06
F6F0 13 AF 12 1B CD A0 F7 04 : 57
F6F8 18 F1 D5 3A 68 F2 5F 3A : 0B
---------------------------------
SUM: 30 B4 71 27 F3 81 99 46 7891

F700 7C F2 BB 38 14 57 3A 6C : 72
F708 F2 83 BA 38 0C 7A 11 70 : 6E
F710 F2 12 13 AF 12 1B CD A0 : 60
F718 F7 3A 69 F2 11 70 F2 12 : 11
F720 13 3E 08 12 1B CD A0 F7 : EA
F728 3A 6D F2 57 3A 69 F2 92 : 17
F730 ED 5B 7A F2 12 D1 C9 C5 : 25
F738 D5 1A FE 80 D2 72 F7 4F : F7
F740 3A 69 F2 81 32 69 F2 4F : F2
F748 3A 6B F2 B9 30 03 32 69 : 1E
F750 F2 3A 69 F2 4F 3A 68 F2 : 6A
F758 47 3A 6C F2 80 B9 D2 9D : 87
F760 F7 3A 6C F2 47 3A 69 F2 : 6B
F768 90 32 68 F2 CD 00 F2 C3 : 9E
F770 9D F7 3A 6D F2 47 3A 69 : 17
F778 F2 4F 1A 81 FE 80 30 08 : 92
---------------------------------
SUM: 29 DB 44 DC B1 35 7F 98 9692

F780 B8 38 05 32 69 F2 18 04 : 9E
F788 78 32 69 F2 3A 68 F2 4F : E8
F790 3A 69 F2 B9 D2 9D F7 32 : E6
F798 68 F2 CD 00 F2 D1 C1 C9 : 74
F7A0 CD E6 F8 CD C0 F7 C9 D5 : CD
F7A8 11 70 F2 3A 69 F2 12 CD : E7
F7B0 E6 F8 3A 69 F2 B7 28 06 : 58
F7B8 11 54 F2 3E 2E 12 D1 C9 : 6F
F7C0 C5 D5 E5 3A 68 F2 4F 1A : 7C
F7C8 91 6F 3A 75 F2 85 6F 3A : CF
F7D0 76 F2 85 6F 26 00 13 1A : AF
F7D8 32 71 F2 3A 6E F2 5F 16 : A4
F7E0 00 CD 68 FA CD 89 FA DD : 5C
F7E8 21 B7 FA 3E 28 DD 77 00 : 8C
F7F0 CD 0A F8 21 40 F2 DD 70 : 6F
F7F8 02 CD 4F F8 DD 46 02 23 : 5E
---------------------------------
SUM: 95 69 82 34 B0 81 16 B3 B71F

F800 03 DD 35 00 20 F0 E1 D1 : D7
F808 C1 C9 AF DD 77 03 DD 77 : E4
F810 04 3A 71 F2 F6 07 DD 77 : F2
F818 01 E6 08 C0 3A 73 F2 FE : 4C
F820 04 3E 06 CA 4B F8 3A 74 : 03
F828 F2 B7 3E 03 C2 4B F8 3A : 29
F830 72 F2 B7 28 05 16 04 DD : 3F
F838 72 03 3A 73 F2 B7 C8 1E : B1
F840 05 FE 03 28 02 1E 04 DD : 2F
F848 73 04 C9 DD 77 01 C9 3A : 98
F850 B6 FA B7 28 10 F3 3E 1D : ED
F858 D3 00 7E CD 99 30 30 29 : 40
F860 3E 1E D3 00 FB DD 7E 00 : 85
F868 DD 5E 01 DD 56 03 FE 13 : 83
```

```
F870 CC D7 F8 FE 12 CC D7 F8 : 46
F878 FE 11 CC D7 F8 FE 08 CC : 7C
---------------------------------
SUM: 89 10 2B A3 48 69 21 9A 9ABE

F880 DD F8 7B 5E 16 00 C3 AF : 36
F888 F8 56 23 5E 2B CD 81 2F : 77
F890 CD B6 2F DD B6 01 CD AF : C2
F898 F8 DD 46 02 03 DD 70 02 : 6F
F8A0 23 DD 35 00 CB F2 CD AF : 6E
F8A8 F8 3E 1E D3 00 FB C9 F5 : E0
F8B0 F5 DD 7E 02 C6 20 47 F1 : 70
F8B8 ED 79 DD 7E 02 C6 38 47 : 08
F8C0 ED 51 DD 7E 02 C6 30 47 : D8
F8C8 ED 59 F1 C9 00 00 00 00 : 00
F8D0 00 00 00 00 00 00 00 7A : 7A
F8D8 B7 C8 5A AF C9 DD 56 04 : 88
F8E0 7A B7 C8 5A AF C9 C5 D5 : 65
F8E8 E5 ED 53 C3 FA 21 40 F2 : 35
F8F0 11 41 F2 01 27 00 3E 20 : CA
F8F8 77 ED B0 ED 5B C3 FA 1A : 33
---------------------------------
SUM: 0F 96 A6 EF 83 CE 59 31 4551

F900 B7 20 06 CD C5 F9 C3 42 : 6D
F908 F9 FE 01 20 09 2A 78 F2 : B5
F910 22 C5 FA C3 34 F9 5F 2A : 5A
F918 C5 FA 3A 6A F2 FE 01 20 : 74
F920 06 2A 78 F2 22 C5 FA 93 : 0E
F928 28 0A 57 DC D4 F9 D4 EA : F0
F930 F9 22 C5 FA 11 47 F2 CD : F1
F938 03 FA ED 5B C3 FA 1A 32 : 4E
F940 6A F2 CD 54 F9 CD 0E FA : 4B
F948 ED 5B C3 FA 1A 32 7C F2 : BF
F950 E1 D1 C1 C9 AF 32 74 F2 : 83
F958 3A 60 F2 FE 44 21 CD FA : B6
F960 28 15 CD 80 F9 D8 ED 5B : A3
F968 C3 FA CD A8 F9 A6 C8 21 : BA
F970 D2 FA 3E 01 32 74 F2 11 : B4
F978 41 F2 01 05 00 ED B0 C9 : 9F
---------------------------------
SUM: 31 A6 D8 80 E8 4A 97 28 0AAE

F980 21 58 FD 11 D8 FD 06 80 : E2
F988 1A BE 20 06 13 23 10 F8 : 3C
F990 B7 C9 78 FE 04 DC 9A F9 : 69
F998 37 C9 11 49 FD 21 48 FD : BD
F9A0 01 0F 00 AF 77 ED B0 C9 : 9C
F9A8 1A CB 3F CB 3F CB 3F 4F : 87
F9B0 06 00 21 48 FD 09 1A E6 : 75
F9B8 07 0E 80 B7 28 05 CB 09 : 4D
F9C0 3D 18 F8 79 C9 11 47 F2 : D9
F9C8 3E 2E 12 13 12 11 60 F2 : 06
F9D0 3E 44 12 C9 F5 7A ED 44 : FD
F9D8 57 01 17 00 09 CD 12 E0 : 37
F9E0 23 FE 0D 20 F8 15 20 F4 : 6F
F9E8 F1 C9 F5 01 19 00 B7 ED : 6D
F9F0 42 CD FA F9 23 15 20 F6 : 50
F9F8 F1 C9 CD 12 E0 FE 0D C8 : 4C
```

```
---------------------------------
SUM: A8 78 82 58 B4 74 76 1C A221

FA00 2B 18 F7 CD 12 E0 FE 0D : 04
FA08 C8 23 12 13 18 F5 AF 32 : FE
FA10 72 F2 32 73 F2 3A 60 F2 : 87
FA18 FE 44 20 06 3E 04 32 73 : 4F
FA20 F2 C9 21 C7 FA CD 47 FA : AB
FA28 21 CA FA CD 47 FA 16 03 : 0C
FA30 3A 60 F2 FE 4D 28 0B 16 : 20
FA38 01 FE 42 28 05 16 02 FE : 84
FA40 41 C0 7A 32 73 F2 C9 C5 : A0
FA48 11 55 F2 06 03 1A 13 CD : 5B
FA50 5F FA BE 23 20 07 10 F5 : 66
FA58 3E 01 32 72 F2 C1 C9 FE : 5D
FA60 61 D8 FE 7B D0 D6 20 C9 : 41
FA68 3A 6F F2 B7 28 CA 7D FA : 02
FA70 29 29 29 29 E5 29 29 19 : F4
FA78 D1 19 44 4D C9 29 29 29 : BF
---------------------------------
SUM: 35 FB 63 CF 1B DE 4D 3F EA72

FA80 E5 29 29 19 D1 19 44 4D : CB
FA88 C9 C5 3A 77 F2 B7 3E 00 : 26
FA90 20 1F 3A 7F D0 B7 28 19 : C0
FA98 F3 3E 1D ED 79 3A 00 00 : EE
FAA0 4F 3A 00 10 47 3E 1E D3 : 0F
FAA8 00 FB 78 B9 3E 01 20 01 : 8C
FAB0 AF 32 B6 FA C1 C9 00 00 : 1B
FAB8 00 00 00 00 00 00 00 00 : 00
FAC0 00 00 00 00 00 00 00 58 : 58
FAC8 31 20 42 41 54 3C 44 49 : F1
FAD0 52 3E 3C 4D 52 4B 3E    : F4
---------------------------------
SUM: 42 10 66 4D F8 50 6A DB 19A7
```

リスト6

```
FE80 00 00 00 00 00 00 00 00 : 00
FE88 00 00 00 00 00 06 00 00 : 06
FE90 00 00 00 00 00 00 00 00 : 00
FE98 00 00 00 00 00 00 04 03 : 07
FEA0 5F 00 00 00 00 00 00 00 : 5F
FEA8 00 00 60 09 00 0A 07 5E : D8
FEB0 00 03 01 00 5C 08 5D 04 : C9
FEB8 02 00 00 00 00 00 00 00 : 02
FEC0 00 5B 5B 5B 5B 5B 5B 00 : 22
FEC8 00 00 00 00 00 00 00 00 : 00
FED0 00 00 00 00 00 00 00 5B : 5B
FED8 5B 5B 00 00 00 00 00 00 : B6
FEE0 00 5B 5B 5B 5B 5B 5B 00 : 22
FEE8 00 00 00 00 00 00 00 00 : 00
FEF0 00 00 00 00 00 00 00 5B : 5B
FEF8 5B 5B 00 00 00 00 00 00 : B6
---------------------------------
SUM: 17 6F 17 BF 12 CE 1E 1B 2888
```

リスト7

```
0000                  1  ; .
0000                  2  ;MENU.X1
0000                  3  ;        WITH KAME-DOS
0000                  4  ;                     Ver 1.0
0000                  5  ;    FOR S-OS REDA
0000                  6
C200                  7   ORG $C200
C200                  8
CD48 P                9  MRKSTK  EQU $CD48
CD58 P               10  NMDIR1  EQU $CD58
CDD8 P               11  NMDIR2  EQU $CDD8
CE58 P               12  POSCRS  EQU $CE58
C200                 13
D080 P               14  #ZOKU   EQU $D080 ;POKE ADR.
D180 P               15  #OD     EQU $D180
D181 P               16  #OP     EQU $D181
D182 P               17  #FCRS   EQU $D182
D184 P               18  #YEN    EQU $D184
D186 P               19  #FSZL   EQU $D186
D18A P               20  #DDRV   EQU $D18A
E054 P               21  #FLNM   EQU $E054
E065 P               22  #SBDR   EQU $E065
E068 P               23  #FNAM   EQU $E068
E078 P               24  #FBYT   EQU $E078
E08C P               25  #STOP   EQU $E08C
E08B P               26  #DN     EQU $E08B
E091 P               27  #BF     EQU $E091
E097 P               28  #MAC    EQU $E097
E09C P               29  #IOFG   EQU $E09C
ECC0 P               30  &MAC4   EQU $ECC0
ECE5 P               31  &ESCP   EQU $ECC0+37
ECDF P               32  #SBUFF  EQU $ECDF
EE80 P               33  P256    EQU $EE00+$80
C200                 34
E003 P               35  #DLFAT  EQU $E003 ;CALL ADR.
E012 P               36  #LDAHL  EQU $E012
E015 P               37  #LDADE  EQU $E015
E018 P               38  #LDDEA  EQU $E018
D006 P               39  #DIR2   EQU $D006
D00C P               40  #DEVI   EQU $D00C
D012 P               41  #DLDIR  EQU $D012
D015 P               42  #SAVED  EQU $D015
D021 P               43  #PREOP  EQU $D021
D024 P               44  #OPENS  EQU $D024
D02A P               45  #DIRSB  EQU $D02A
2F81 P               46  SFTJIS@ EQU $2F81
2FB6 P               47  JISVRM@ EQU $2FB6
3099 P               48  SFTCHK@ EQU $3099
C200                 49  ;
C200 C3 DA C6        50  DIR     JP DIRS
C203 C3 00 C7        51  CRPRT   JP CRPRTS
C206 C3 3D C7        52  LIMIT   JP LIMITS ;USR
C209 C3 AD C7        53  FNWRK   JP FNWRKS
C20C C3 97 C5        54  WIDOW   JP WIDOWS ;
C20F C3 58 C6        55  UNDOW   JP UNDOWS
C212 C3 6C C5        56  MARK    JP MARKS
C215 C3 06 C5        57  CHECK   JP CHECKS
C218 C3 6D C4        58  OPEN    JP OPEN1  ;
C21B C3 DA C3        59  TWOSTP  JP TWOSTS
C21E C3 BE C3        60  CLRWK   JP CLRWKS
C221 C3 35 C5        61  MACON   JP ALLONS
C224 C3 95 C9        62  MACOF   JP MRKCLR
C227 C3 76 C3        63  MARK?   JP MARKS?
C22A C3 51 C3        64  DEL     JP DELS
C22D C3 89 C2        65  COPY    JP COPYS
C230                 66  ;
C230                 67
C240                 68   ORG $C240
C240                 69
C240                 70  WORK    DS 40
C268 00              71  DU      DB 0
C269 00              72  CR      DB 0
C26A 00              73  CB      DB 0
C26B 00              74  DDE     DB 0
C26C 14              75  YS      DB 20
C26D 01              76  UU      DB 1
C26E 00              77  XX      DB 0
C26F 50              78  WIDTH   DB 80
C270                 79  WWWF    DS 5   ;n,ATR
C275 03              80  USP     DB 3
C276 00              81  YY      DB 0
C277 00              82  KMODE   DB 0
C278 00 00           83  NWADR   DW 0
C27A 00 00           84  CBADR   DW 0
C27C 00              85  CB2     DB 0
C27D                 86
C27D 00              87  MAC1    DB 0
C27E 00              88  MAC2    DB 0
C27F                 89  FBXFS   DS 4
C283                 90  UUSR    DS 4
C287 00              91  DSHU    DB 0
C288 00              92  ESCW    DB 0
C289                 93
C289                 94  ;
C289                 95  ;COPY
C289                 96  ;
C289                 97  COPYS
C289 AF              98   XOR A
C28A 32 87 C2        99   LD (DSHU),A
C28D 3E 10          100   LD A,$10
C28F 32 88 C2       101   LD (ESCW),A
C292                102
C292 3A 97 E0       103   LD A,(#MAC)
C295 32 7D C2       104   LD (MAC1),A
C298 3E 07          105   LD A,7
C29A 32 8B D1       106   LD (#DDRV+1),A
C29D 32 8C D1       107   LD (#DDRV+2),A
C2A0 2A 78 E0       108   LD HL,(#FBYT)
C2A3 22 7F C2       109   LD (FBXFS),HL
C2A6 2A 7A E0       110   LD HL,(#FBYT+2)
C2A9 22 81 C2       111   LD (FBXFS+2),HL
C2AC 3E 03          112   LD A,3
C2AE 32 83 C2       113   LD (UUSR),A
C2B1 3E 02          114   LD A,2
C2B3 32 84 C2       115   LD (UUSR+1),A
C2B6 3E 01          116   LD A,1
C2B8 32 85 C2       117   LD (UUSR+2),A
C2BB 3E 03          118   LD A,3
C2BD 32 86 C2       119   LD (UUSR+3),A
C2C0 11 83 C2       120   LD DE,UUSR
C2C3 06 04          121   LD B,4
C2C5 CD 18 C2       122   CALL OPEN
C2C8 3A 8C E0       123   LD A,(#STOP)
C2CB B7             124   OR A
C2CC C0             125   RET NZ
C2CD 3A 97 E0       126   LD A,(#MAC)
C2D0 32 7E C2       127   LD (MAC2),A
C2D3                128
C2D3 CD 0B C3       129   CALL BYMAC
C2D6 CD 22 C3       130   CALL BYMAC2
C2D9 3A 88 C2       131   LD A,(ESCW)
C2DC 32 E5 EC       132   LD (&ESCP),A
C2DF AF             133   XOR A
C2E0 32 9C E0       134   LD (#IOFG),A
C2E3                135  COPYLP
C2E3 CD 0C D0       136     CALL #DEVI
C2E6 3A 8C E0       137     LD A,(#STOP)
C2E9 B7             138     OR A
C2EA C0             139     RET NZ
C2EB 3A 9C E0       140     LD A,(#IOFG)
C2EE B7             141     OR A
C2EF 20 F2          142     JR NZ,COPYLP
C2F1                143
C2F1 CD 39 C3       144   CALL CLFSZ
C2F4 3A 81 D0       145   LD A,(#ZOKU+1)
C2F7 32 82 D0       146   LD (#ZOKU+2),A
C2FA 3A 87 C2       147   LD A,(DSHU)
C2FD B7             148   OR A
C2FE 28 03          149   JR Z,ZOKUSK
C300 32 82 D0       150     LD (#ZOKU+2),A
C303                151  ZOKUSK
C303 3E 02          152   LD A,2
C305 32 80 D1       153   LD (#OD),A
C308 C3 15 D0       154   JP #SAVED
C30B                155
C30B                156  BYMAC
C30B 3A 7D C2       157   LD A,(MAC1)
C30E E6 01          158   AND 1
C310 C0             159   RET NZ
C311 3A 7E C2       160   LD A,(MAC2)
C314 E6 01          161   AND 1
C316 C8             162   RET Z
C317 3E 04          163   LD A,4
C319 32 87 C2       164   LD (DSHU),A
C31C 3E 12          165   LD A,$12
C31E 32 88 C2       166   LD (ESCW),A
C321 C9             167   RET
C322                168
C322                169  BYMAC2
C322 3A 7D C2       170   LD A,(MAC1)
C325 E6 01          171   AND 1
C327 C8             172   RET Z
C328 3A 7E C2       173   LD A,(MAC2)
C32B E6 01          174   AND 1
C32D C0             175   RET NZ
C32E 3E 20          176   LD A,$20
C330 32 87 C2       177   LD (DSHU),A
C333 3E 13          178   LD A,$13
C335 32 88 C2       179   LD (ESCW),A
C338 C9             180   RET
```

▶現在試験期間中です。今日の数学の試験でワイルドカードを使ってみました。結果が楽しみだ。

橘　正彦(17)福島県

```
C339                181
C339                182  CLFSZ
C339 3A E5 EC       183   LD A,(&ESCP)
C33C B7             184   OR A
C33D C0             185   RET NZ
C33E 3A 87 C2       186   LD A,(DSHU)
C341 FE 20          187   CP $20
C343 C8             188   RET Z
C344 2A 7F C2       189   LD HL,(FBXFS)
C347 22 86 D1       190   LD (#FSZL),HL
C34A 2A 81 C2       191   LD HL,(FBXFS+2)
C34D 22 88 D1       192   LD (#FSZL+2),HL
C350 C9             193   RET
C351                194
C351                195  ;
C351                196  ;DEL
C351                197  ;
C351                198  DELS
C351 CD 03 E0       199   CALL #DLFAT
C354 CD 12 D0       200   CALL #DLDIR
C357 CD 15 C2       201   CALL CHECK
C35A 3A 69 C2       202   LD A,(CR)
C35D FE 80          203   CP 128
C35F D0             204   RET NC
C360 11 54 E0       205   LD DE,#FLNM
C363 21 47 C2       206   LD HL,WORK+7
C366 01 0D 00       207   LD BC,13
C369 ED B0          208   LDIR
C36B 21 55 C2       209   LD HL,WORK+21
C36E 01 03 00       210   LD BC,3
C371 ED B0          211   LDIR
C373 C3 06 D0       212   JP #DIR2
C376                213
C376                214  ;
C376                215  ;MARK?
C376                216  ;
C376                217  MARKS?
C376 3A 6E C2       218   LD A,(XX)
C379 C6 04          219   ADD A,4
C37B 5F             220   LD E,A
C37C 16 00          221   LD D,0
C37E 3A 76 C2       222   LD A,(YY)
C381 21 75 C2       223   LD HL,USP
C384 86             224   ADD A,(HL)
C385 21 6C C2       225   LD HL,YS
C388 86             226   ADD A,(HL)
C389 3C             227   INC A
C38A 6F             228   LD L,A
C38B 26 00          229   LD H,0
C38D CD 63 CA       230   CALL XYADR
C390                231
C390 C5             232   PUSH BC
C391 06 08          233   LD B,8
C393 21 48 CD       234   LD HL,MRKSTK
C396                235  MARKSLP
C396 7E             236     LD A,(HL)
C397 23             237     INC HL
C398 B7             238     OR A
C399 20 06          239     JR NZ,MARSK?
C39B 10 F9          240     DJNZ MARKSLP
C39D 16 20          241   LD D," "
C39F 18 02          242   JR MARSKS?
C3A1                243  MARSK?
C3A1 16 4D          244   LD D,"M"
C3A3                245  MARSKS?
C3A3 7A             246   LD A,D
C3A4 32 84 D1       247   LD (#YEN),A
C3A7 C1             248   POP BC
C3A8 3E 20          249   LD A,$20
C3AA 80             250   ADD A,B
C3AB 47             251   LD B,A
C3AC 3E 03          252   LD A,3
C3AE ED 79          253   OUT (C),A
C3B0 3E 18          254   LD A,$18
C3B2 80             255   ADD A,B
C3B3 47             256   LD B,A
C3B4 AF             257   XOR A
C3B5 ED 79          258   OUT (C),A
C3B7 78             259   LD A,B
C3B8 D6 08          260   SUB 8
C3BA 47             261   LD B,A
C3BB ED 51          262   OUT (C),D
C3BD C9             263   RET
C3BE                264
C3BE                265  ;
C3BE                266  ;CLEAR WORKS
C3BE                267  ;
C3BE                268  CLRWKS
C3BE 11 49 CD       269   LD DE,MRKSTK+1
C3C1 21 48 CD       270   LD HL,MRKSTK
C3C4 01 37 01       271   LD BC,15+256+40
C3C7 AF             272   XOR A
C3C8 77             273   LD (HL),A
C3C9 ED B0          274   LDIR
C3CB C9             275   RET
C3CC                276
C3CC                277  CLRMRK
C3CC 11 49 CD       278   LD DE,MRKSTK+1
C3CF 21 48 CD       279   LD HL,MRKSTK
C3D2 01 0F 00       280   LD BC,15
C3D5 AF             281   XOR A
C3D6 77             282   LD (HL),A
C3D7 ED B0          283   LDIR
C3D9 C9             284   RET
C3DA                285
C3DA                286  ;
C3DA                287  ;OPEN*2
C3DA                288  ;
C3DA                289  TWOSTS
C3DA DD 21 80 EE    290   LD IX,P256
C3DE 3E 01          291   LD A,1
C3E0 DD 77 00       292   LD (IX+0),A
C3E3 DD 77 01       293   LD (IX+1),A
C3E6 DD 77 03       294   LD (IX+3),A
C3E9 AF             295   XOR A
C3EA DD 77 02       296   LD (IX+2),A
C3ED 11 84 EE       297   LD DE,P256+4
C3F0 21 D8 CD       298   LD HL,NMDIR2
C3F3 CD 63 C4       299   CALL PDEHL
C3F6 21 3C CE       300   LD HL,NMDIR2+100
C3F9 CD 63 C4       301   CALL PDEHL
C3FC CD 58 C4       302   CALL TWOPN
C3FF 3A 8C E0       303   LD A,(#STOP)
C402 B7             304   OR A
C403 C2 CC C3       305   JP NZ,CLRMRK
C406                306
C406 CD 2A D0       307   CALL #DIRSB
C409 3A 84 D1       308   LD A,(#YEN)
C40C 32 6B C2       309   LD (DDE),A
C40F 2A 91 E0       310   LD HL,(#BF)
C412 22 78 C2       311   LD (NWADR),HL
C415 CD 15 C2       312   CALL CHECK
C418 CD 48 C4       313   CALL DSKIDF
C41B 3A 69 C2       314   LD A,(CR)
C41E FE 80          315   CP 128
C420 D0             316   RET NC
C421 DD 21 80 EE    317   LD IX,P256
C425 3E 01          318   LD A,1
C427 DD 77 00       319   LD (IX+0),A
C42A DD 77 01       320   LD (IX+1),A
C42D DD 77 02       321   LD (IX+2),A
C430 AF             322   XOR A
C431 DD 77 03       323   LD (IX+3),A
C434 11 84 EE       324   LD DE,P256+4
C437 21 D8 CD       325   LD HL,NMDIR2
C43A CD 63 C4       326   CALL PDEHL
C43D 21 47 C2       327   LD HL,WORK+7
C440 01 11 00       328   LD BC,17
C443 ED B0          329   LDIR
C445 C3 58 C4       330   JP TWOPN
C448                331
C448                332  DSKIDF
C448 3A 84 D1       333   LD A,(#YEN)
C44B 21 55 CE       334   LD HL,NMDIR2+125
C44E BE             335   CP (HL)
C44F C8             336   RET Z
C450 3E 80          337   LD A,128
C452 32 69 C2       338   LD (CR),A
C455 C3 CC C3       339   JP CLRMRK
C458                340
C458                341  ;
C458                342  TWOPN
C458 EB             343   EX DE,HL
C459 11 80 EE       344   LD DE,P256
C45C B7             345   OR A
C45D ED 52          346   SBC HL,DE
C45F 45             347   LD B,L
C460 C3 18 C2       348   JP OPEN
C463                349
C463                350  PDEHL
C463 7E             351   LD A,(HL)
C464 B7             352   OR A
C465 C8             353   RET Z
C466 4F             354   LD C,A
C467 06 00          355   LD B,0
C469 23             356   INC HL
C46A ED B0          357   LDIR
C46C C9             358   RET
C46D                359
C46D                360  ;
C46D                361  ;"OPEN"
C46D                362  ;
C46D                363  OPEN1
C46D 1A             364   LD A,(DE)
C46E 13             365   INC DE
C46F 05             366   DEC B
C470 FE 02          367   CP 2
C472 28 25          368   JR Z,OPEN2
C474 FE 03          369   CP 3
C476 28 2A          370   JR Z,OPEN3
C478                371
C478 CD C2 C4       372   CALL SMACDN
C47B CD D3 C4       373   CALL ODSBOP
C47E C5             374   PUSH BC
C47F D5             375   PUSH DE
C480 CD 21 D0       376   CALL #PREOP
C483 D1             377   POP DE
C484 C1             378   POP BC
C485 3A 8C E0       379   LD A,(#STOP)
C488 B7             380   OR A
C489 C0             381   RET NZ
C48A 68             382   LD L,B
C48B 3A 84 D1       383   LD A,(#YEN)
C48E 47             384   LD B,A
C48F 7D             385   LD A,L
C490 90             386   SUB B
C491 6F             387   LD L,A
C492 26 00          388   LD H,0
C494 19             389   ADD HL,DE
C495 EB             390   EX DE,HL
C496 C3 24 D0       391   JP #OPENS
C499                392
C499                393  OPEN2
C499 1A             394   LD A,(DE)
C49A 13             395   INC DE
C49B 05             396   DEC B
C49C 32 80 D1       397   LD (#OD),A
C49F C3 21 D0       398   JP #PREOP
C4A2                399
C4A2                400  OPEN3
C4A2 CD C2 C4       401   CALL SMACDN
C4A5 CD E6 C4       402   CALL STKFN
C4A8 CD D3 C4       403   CALL ODSBOP
C4AB 3A 58 CD       404   LD A,(NMDIR1)
C4AE 47             405   LD B,A
C4AF 11 59 CD       406   LD DE,NMDIR1+1
C4B2 CD 21 D0       407   CALL #PREOP
C4B5 CD F6 C4       408   CALL LODFN
C4B8 3A 8C E0       409   LD A,(#STOP)
C4BB B7             410   OR A
C4BC C0             411   RET NZ
C4BD 06 00          412   LD B,0
C4BF C3 24 D0       413   JP #OPENS
C4C2                414
C4C2                415  ;
C4C2                416  SMACDN
C4C2 C5             417   PUSH BC
C4C3 3A 8B E0       418   LD A,(#DN)
C4C6 4F             419   LD C,A
C4C7 06 00          420   LD B,0
C4C9 21 C0 EC       421   LD HL,&MAC4
C4CC 09             422   ADD HL,BC
C4CD 7E             423   LD A,(HL)
C4CE 32 97 E0       424   LD (#MAC),A
C4D1 C1             425   POP BC
C4D2 C9             426   RET
C4D3                427
C4D3                428  ODSBOP
C4D3 1A             429   LD A,(DE)
C4D4 32 80 D1       430   LD (#OD),A
C4D7 13             431   INC DE
C4D8 05             432   DEC B
C4D9 1A             433   LD A,(DE)
C4DA 32 65 E0       434   LD (#SBDR),A
C4DD 13             435   INC DE
C4DE 05             436   DEC B
C4DF 1A             437   LD A,(DE)
C4E0 32 81 D1       438   LD (#OP),A
C4E3 13             439   INC DE
C4E4 05             440   DEC B
C4E5 C9             441   RET
C4E6                442
C4E6                443  STKFN
C4E6 C5             444   PUSH BC
C4E7 D5             445   PUSH DE
C4E8 21 68 E0       446   LD HL,#FNAM
C4EB 11 DD EE       447   LD DE,P256+$5D
C4EE 01 23 00       448   LD BC,35
C4F1 ED B0          449   LDIR
C4F3 D1             450   POP DE
C4F4 C1             451   POP BC
C4F5 C9             452   RET
C4F6                453
C4F6                454  LODFN
C4F6 C5             455   PUSH BC
C4F7 D5             456   PUSH DE
C4F8 11 68 E0       457   LD DE,#FNAM
C4FB 21 DD EE       458   LD HL,P256+$5D
C4FE 01 23 00       459   LD BC,35
C501 ED B0          460   LDIR
C503 D1             461   POP DE
C504 C1             462   POP BC
C505 C9             463   RET
C506                464
C506                465  ;
C506                466  ;MARK CHECK
C506                467  ;
C506                468  CHECKS ;CF
C506 11 69 C2       469   LD DE,CR
C509                470  CHKLP
C509 1A             471     LD A,(DE)
C50A FE 01          472     CP 1
C50C 20 09          473     JR NZ,CHKLPSK
C50E 32 6A C2       474       LD (CB),A
C511 2A 78 C2       475       LD HL,(NWADR)
C514 22 C0 CA       476       LD (WTOPA),HL
C517                477  CHKLPSK
C517 CD A3 C9       478     CALL RRCA
C51A A6             479     AND (HL)
C51B 20 0B          480     JR NZ,CHKSK1
C51D 1A             481     LD A,(DE) •
C51E 21 55 CE       482     LD HL,NMDIR2+125
C521 BE             483     CP (HL)
C522 30 0C          484     JR NC,CHKSK2
C524 3C             485     INC A
C525 12             486     LD (DE),A
C526 18 E1          487     JR CHKLP
C528                488  CHKSK1
C528 2F             489   CPL
C529 A6             490   AND (HL)
C52A 77             491   LD (HL),A
C52B CD 09 C2       492   CALL FNWRK
C52E B7             493   OR A
C52F C9             494   RET
C530                495
C530                496  CHKSK2
C530 3E 80          497   LD A,128
C532 12             498   LD (DE),A
C533 37             499   SCF
C534 C9             500   RET
C535                501
C535                502  ;
C535                503  ;ALL MARK
C535                504  ;
C535                505  ALLONS
C535 CD 7B C9       506   CALL DIRIDF
C538 30 0E          507   JR NC,ALLSSB
C53A                508
C53A 11 D8 CD       509   LD DE,NMDIR2
C53D 21 58 CD       510   LD HL,NMDIR1
C540 01 80 00       511   LD BC,128
C543 ED B0          512   LDIR
C545 CD 95 C9       513   CALL MRKCLR
C548                514
C548                515  ALLSSB
C548 3E 01          516   LD A,1
C54A 32 69 C2       517   LD (CR),A
C54D 3A 55 CE       518   LD A,(NMDIR2+125)
C550 47             519   LD B,A
C551                520  ALLLP
C551 C5             521     PUSH BC
C552 CD 09 C2       522     CALL FNWRK
C555 3A 73 C2       523     LD A,(WWWF+3)
C558 FE 04          524     CP 4
C55A 28 08          525     JR Z,ALLSK
C55C 11 69 C2       526     LD DE,CR
C55F CD A3 C9       527     CALL RRCA
C562 B6             528     OR (HL)
C563 77             529     LD (HL),A
C564                530  ALLSK
C564 21 69 C2       531     LD HL,CR
C567 34             532     INC (HL)
C568 C1             533     POP BC
C569 10 E6          534     DJNZ ALLLP
C56B C9             535   RET
C56C                536
C56C                537  ;
C56C                538  ;MARK
C56C                539  ;
C56C                540  MARKS
C56C 3A 73 C2       541   LD A,(WWWF+3)
C56F FE 04          542   CP 4
C571 C8             543   RET Z
C572 CD 7B C9       544   CALL DIRIDF
C575 30 0E          545   JR NC,MARKSSB
C577                546
C577 11 D8 CD       547   LD DE,NMDIR2
C57A 21 58 CD       548   LD HL,NMDIR1
C57D 01 80 00       549   LD BC,128
C580 ED B0          550   LDIR
C582 CD 95 C9       551   CALL MRKCLR
C585                552
C585                553  MARKSSB
C585 11 69 C2       554   LD DE,CR
C588 CD A3 C9       555   CALL RRCA
C58B 57             556   LD D,A
C58C A6             557   AND (HL)
C58D 20 04          558   JR NZ,MARKSSK
C58F 7A             559     LD A,D
C590 B6             560     OR (HL)
C591 77             561     LD (HL),A
C592 C9             562     RET
C593                563  MARKSSK
C593 2F             564   CPL
C594 A6             565   AND (HL)
C595 77             566   LD (HL),A
C596 C9             567   RET
C597                568
C597                569  ;
C597                570  ;WINDOW
C597                571  ;
C597                572  WIDOWS
C597 D5             573   PUSH DE
C598 DD E1          574   POP IX
C59A 3A 6E C2       575   LD A,(XX)
C59D DD 86 00       576   ADD A,(IX+0)
C5A0 5F             577   LD E,A
C5A1 16 00          578   LD D,0
C5A3 3A 76 C2       579   LD A,(YY)
C5A6 DD 86 01       580   ADD A,(IX+1)
C5A9 6F             581   LD L,A
C5AA 26 00          582   LD H,0
C5AC CD 63 CA       583   CALL XYADR
C5AF ED 43 B7 CA    584   LD (WXYSS),BC
C5B3 DD 7E 02       585   LD A,(IX+2)
C5B6 32 B9 CA       586   LD (WXYSS+2),A
C5B9 DD 7E 03       587   LD A,(IX+3)
C5BC 32 BA CA       588   LD (WXYSS+3),A
C5BF ED 5B DF EC    589   LD DE,(#SBUFF)
C5C3 AF             590   XOR A
C5C4 21 00 38       591   LD HL,$3800
C5C7 CD DB C5       592   CALL WNDSB ;X1ではこの
C5CA                593              ;3バイトなし
C5CA 3E 20          594   LD A,$20
C5CC 21 20 30       595   LD HL,$3020
C5CF CD DB C5       596   CALL WNDSB
C5D2 3E 07          597   LD A,7
C5D4 21 0F 20       598   LD HL,$200F
C5D7 CD DB C5       599   CALL WNDSB
C5DA C9             600   RET
```

▶学生のころはPC-9801が買いたかった。会社でFM-Rを使い出した。なにかが，なにかが違う！　で，X68000になったのである。セグメントのない世界はいいなあ。
和田　岳雄(21)福岡県

```
C5DB                601
C5DB                602  WNDSB
C5DB C5             603   PUSH BC
C5DC 32 BD CA       604   LD (WXYSS+6),A
C5DF 78             605   LD A,B
C5E0 84             606   ADD A,H
C5E1 47             607   LD B,A
C5E2 32 BC CA       608   LD (WXYSS+5),A
C5E5 CD 2E C6       609   CALL WNDSB2
C5E8 CD ED C5       610   CALL KAGE
C5EB C1             611   POP BC
C5EC C9             612   RET
C5ED                613
C5ED                614  KAGE
C5ED 03             615   INC BC
C5EE DD 6E 02       616   LD L,(IX+2)
C5F1                617  KAGELPY
C5F1 ED 78          618      IN A,(C)
C5F3 CD 18 E0       619      CALL #LDDEA
C5F6 13             620      INC DE
C5F7 3A BD CA       621      LD A,(WXYSS+6)
C5FA ED 79          622      OUT (C),A
C5FC 03             623      INC BC
C5FD 2D             624      DEC L
C5FE 20 F1          625      JR NZ,KAGELPY
C600 ED 4B B7 CA    626   LD BC,(WXYSS)
C604 3A BC CA       627   LD A,(WXYSS+5)
C607 47             628   LD B,A
C608 DD 7E 02       629   LD A,(IX+2)
C60B 6F             630   LD L,A
C60C 26 00          631   LD H,0
C60E 09             632   ADD HL,BC
C60F 44             633   LD B,H
C610 4D             634   LD C,L
C611 DD 6E 03       635   LD L,(IX+3)
C614                636  KAGELPT
C614 E5             637      PUSH HL
C615 3A 6F C2       638      LD A,(WIDTH)
C618 6F             639      LD L,A
C619 26 00          640      LD H,0
C61B 09             641      ADD HL,BC
C61C 44             642      LD B,H
C61D 4D             643      LD C,L
C61E E1             644      POP HL
C61F 2D             645      DEC L
C620 C8             646      RET Z
C621 ED 78          647      IN A,(C)
C623 CD 18 E0       648      CALL #LDDEA
C626 13             649      INC DE
C627 3A BD CA       650      LD A,(WXYSS+6)
C62A ED 79          651      OUT (C),A
C62C 18 E6          652      JR KAGELPT
C62E                653
C62E                654  WNDSB2
C62E 7D             655   LD A,L
C62F 32 BB CA       656   LD (WXYSS+4),A
C632 DD 6E 03       657   LD L,(IX+3)
C635                658  WNDLP1
C635 DD 66 02       659      LD H,(IX+2)
C638 C5             660      PUSH BC
C639                661  WNDLP2
C639 ED 78          662         IN A,(C)
C63B CD 18 E0       663         CALL #LDDEA
C63E 13             664         INC DE
C63F 3A BB CA       665         LD A,(WXYSS+4)
C642 ED 79          666         OUT (C),A
C644 03             667         INC BC
C645 25             668         DEC H
C646 20 F1          669         JR NZ,WNDLP2
C648 C1             670      POP BC
C649 E5             671      PUSH HL
C64A 3A 6F C2       672      LD A,(WIDTH)
C64D 6F             673      LD L,A
C64E 26 00          674      LD H,0
C650 09             675      ADD HL,BC
C651 44             676      LD B,H
C652 4D             677      LD C,L
C653 E1             678      POP HL
C654 2D             679      DEC L
C655 20 DE          680      JR NZ,WNDLP1
C657 C9             681   RET
C658                682
C658                683  ;
C658                684  ;WINDOW BACK
C658                685  ;
C658                686  UNDOWS
C658 ED 4B B7 CA    687   LD BC,(WXYSS)
C65C ED 5B DF EC    688   LD DE,(#SBUFF)
C660 3E 38          689   LD A,$38
C662 CD 70 C6       690   CALL UNDSBB ;X1ではこの
C665                691               ;3バイトなし
C665 3E 30          692   LD A,$30
C667 CD 70 C6       693   CALL UNDSBB
C66A 3E 20          694   LD A,$20
C66C CD 70 C6       695   CALL UNDSBB
C66F C9             696   RET
C670                697
C670                698  UNDSBB
C670 C5             699   PUSH BC
C671 80             700   ADD A,B
C672 47             701   LD B,A
C673 32 BC CA       702   LD (WXYSS+5),A
C676 CD B7 C6       703   CALL UNDSBB2
C679 CD 7E C6       704   CALL UNKAGE
C67C C1             705   POP BC
C67D C9             706   RET
C67E                707
C67E                708  UNKAGE
C67E 03             709   INC BC
C67F 3A B9 CA       710   LD A,(WXYSS+2)
C682 6F             711   LD L,A
C683                712  UNKGLPY
C683 CD 15 E0       713      CALL #LDADE
C686 13             714      INC DE
C687 ED 79          715      OUT (C),A
C689 03             716      INC BC
C68A 2D             717      DEC L
C68B 20 F6          718      JR NZ,UNKGLPY
C68D ED 4B B7 CA    719   LD BC,(WXYSS)
C691 3A BC CA       720   LD A,(WXYSS+5)
C694 47             721   LD B,A
C695 3A B9 CA       722   LD A,(WXYSS+2)
C698 6F             723   LD L,A
C699 26 00          724   LD H,0
C69B 09             725   ADD HL,BC
C69C 44             726   LD B,H
C69D 4D             727   LD C,L
C69E 3A BA CA       728   LD A,(WXYSS+3)
C6A1 6F             729   LD L,A
C6A2                730  UNKGLPT
C6A2 E5             731      PUSH HL
C6A3 3A 6F C2       732      LD A,(WIDTH)
C6A6 6F             733      LD L,A
C6A7 26 00          734      LD H,0
C6A9 09             735      ADD HL,BC
C6AA 44             736      LD B,H
C6AB 4D             737      LD C,L
C6AC E1             738      POP HL
C6AD 2D             739      DEC L
C6AE C8             740      RET Z
C6AF CD 15 E0       741      CALL #LDADE
C6B2 13             742      INC DE
C6B3 ED 79          743      OUT (C),A
C6B5 18 EB          744      JR UNKGLPT
C6B7                745
C6B7                746  UNDSBB2
C6B7 3A BA CA       747   LD A,(WXYSS+3)
C6BA 6F             748   LD L,A
C6BB                749  UNDLP1
C6BB 3A B9 CA       750      LD A,(WXYSS+2)
C6BE 67             751      LD H,A
C6BF C5             752      PUSH BC
C6C0                753  UNDLP2
C6C0 CD 15 E0       754         CALL #LDADE
C6C3 13             755         INC DE
C6C4 ED 79          756         OUT (C),A
C6C6 03             757         INC BC
C6C7 25             758         DEC H
C6C8 20 F6          759         JR NZ,UNDLP2
C6CA C1             760      POP BC
C6CB E5             761      PUSH HL
C6CC 3A 6F C2       762      LD A,(WIDTH)
C6CF 6F             763      LD L,A
C6D0 26 00          764      LD H,0
C6D2 09             765      ADD HL,BC
C6D3 44             766      LD B,H
C6D4 4D             767      LD C,L
C6D5 E1             768      POP HL
C6D6 2D             769      DEC L
C6D7 20 E2          770      JR NZ,UNDLP1
C6D9 C9             771   RET
C6DA                772
C6DA                773  ;
C6DA                774  ;FOR DU TO K
C6DA                775  ;
C6DA                776  DIRS
C6DA 3A 68 C2       777   LD A,(DU)
C6DD 4F             778   LD C,A
C6DE 3A 6C C2       779   LD A,(YS)
C6E1 81             780   ADD A,C
C6E2 4F             781   LD C,A
C6E3 3A 6B C2       782   LD A,(DDE)
C6E6 B9             783   CP C
C6E7 30 01          784   JR NC,DIRSSK
C6E9 4F             785     LD C,A
C6EA                786  DIRSSK
C6EA 3A 68 C2       787   LD A,(DU)
C6ED 47             788   LD B,A
C6EE 11 70 C2       789   LD DE,WWWF
C6F1                790  DIRSLP
C6F1 79             791      LD A,C
C6F2 B8             792      CP B
C6F3 D8             793      RET C
C6F4 78             794      LD A,B
C6F5 12             795      LD (DE),A
C6F6 13             796      INC DE
C6F7 AF             797      XOR A
C6F8 12             798      LD (DE),A
C6F9 1B             799      DEC DE
C6FA CD A6 C7       800      CALL FNPRTS
C6FD 04             801      INC B
C6FE 18 F1          802      JR DIRSLP
C700                803
C700                804  ;
C700                805  ;CREV:SCRN
C700                806  ;
C700                807  CRPRTS
C700 D5             808   PUSH DE
C701 3A 68 C2       809     LD A,(DU)
C704 5F             810     LD E,A
C705 3A 7C C2       811     LD A,(CB2)
C708 BB             812     CP E ;IF CB<DU
C709 38 14          813     JR C,CRPSK
C70B 57             814     LD D,A
C70C 3A 6C C2       815     LD A,(YS)
C70F 83             816     ADD A,E
C710 BA             817     CP D ;IF DU+YS<CB
C711 38 0C          818     JR C,CRPSK
C713 7A             819   LD A,D
C714 11 70 C2       820   LD DE,WWWF
C717 12             821   LD (DE),A
C718 13             822   INC DE
C719 AF             823   XOR A
C71A 12             824   LD (DE),A
C71B 1B             825   DEC DE
C71C CD A6 C7       826   CALL FNPRTS
C71F                827  CRPSK
C71F 3A 69 C2       828   LD A,(CR)
C722 11 70 C2       829   LD DE,WWWF
C725 12             830   LD (DE),A
C726 13             831   INC DE
C727 3E 08          832   LD A,8
C729 12             833   LD (DE),A ;CREV 1
C72A 1B             834   DEC DE
C72B CD A6 C7       835   CALL FNPRTS
C72E 3A 6D C2       836   LD A,(UU)
C731 57             837   LD D,A
C732 3A 69 C2       838   LD A,(CR)
C735 92             839   SUB D
C736 ED 5B 7A C2    840   LD DE,(CBADR)
C73A 12             841   LD (DE),A
C73B D1             842   POP DE
C73C C9             843   RET
C73D                844
C73D                845  ;
C73D                846  ;LIMIT
C73D                847  ;
C73D                848  LIMITS
C73D C5             849   PUSH BC
C73E D5             850   PUSH DE
C73F 1A             851   LD A,(DE)
C740 FE 80          852   CP $80
C742 D2 78 C7       853   JP NC,UPLMT
C745                854
C745 4F             855   LD C,A
C746 3A 69 C2       856   LD A,(CR)
C749 81             857   ADD A,C
C74A 32 69 C2       858   LD (CR),A
C74D 4F             859   LD C,A
C74E 3A 6B C2       860   LD A,(DDE)
C751 B9             861   CP C ;IF DE<CR
C752 30 03          862   JR NC,LMTSK
C754 32 69 C2       863     LD (CR),A
C757                864  LMTSK
C757 3A 69 C2       865   LD A,(CR)
C75A 4F             866   LD C,A
C75B 3A 68 C2       867   LD A,(DU)
C75E 47             868   LD B,A
C75F 3A 6C C2       869   LD A,(YS)
C762 80             870   ADD A,B
C763 B9             871   CP C ;IF DU+YS=>CR
C764 D2 A3 C7       872   JP NC,LMTQ
C767 3A 6C C2       873   LD A,(YS)
C76A 47             874   LD B,A
C76B 3A 69 C2       875   LD A,(CR)
C76E 90             876   SUB B
C76F 32 68 C2       877   LD (DU),A
C772 CD 00 C2       878   CALL DIR
C775 C3 A3 C7       879   JP LMTQ
C778                880
C778                881  UPLMT
C778 3A 6D C2       882   LD A,(UU)
C77B 47             883   LD B,A
C77C 3A 69 C2       884   LD A,(CR)
C77F 4F             885   LD C,A
C780 1A             886   LD A,(DE) ;K
C781 81             887   ADD A,C
C782 FE 80          888   CP $80
C784 30 08          889   JR NC,UPLMTSK1
C786 B8             890   CP B
C787 38 05          891   JR C,UPLMTSK1
C789 32 69 C2       892   LD (CR),A ;CR=CR+K
C78C 18 04          893   JR UPLMTSK2
C78E                894  UPLMTSK1
C78E 78             895     LD A,B  ;CR=UU
C78F 32 69 C2       896     LD (CR),A
C792                897  UPLMTSK2
C792 3A 68 C2       898   LD A,(DU)
C795 4F             899   LD C,A
C796 3A 69 C2       900   LD A,(CR)
C799 B9             901   CP C
C79A D2 A3 C7       902   JP NC,LMTQ
C79D 32 68 C2       903   LD (DU),A
C7A0 CD 00 C2       904   CALL DIR
C7A3                905
C7A3                906  LMTQ
C7A3 D1             907   POP DE
C7A4 C1             908   POP BC
C7A5 C9             909   RET
C7A6                910
C7A6                911  ;
C7A6                912  ;PRT MEM(WORK)
C7A6                913  ;
C7A6                914  FNPRTS
C7A6 CD E1 C8       915   CALL FNMAKE
C7A9 CD C6 C7       916   CALL FNP
C7AC C9             917   RET
C7AD                918
C7AD                919  ;
C7AD                920  ;FN ->WORK
C7AD                921  ;
C7AD                922  FNWRKS
C7AD D5             923   PUSH DE
C7AE 11 70 C2       924   LD DE,WWWF
C7B1 3A 69 C2       925   LD A,(CR)
C7B4 12             926   LD (DE),A
C7B5 CD E1 C8       927   CALL FNMAKE
C7B8 3A 69 C2       928   LD A,(CR)
C7BB B7             929   OR A
C7BC 28 06          930   JR Z,FNWRKQ
C7BE 11 54 C2       931   LD DE,WORK+7+13
C7C1 3E 2E          932   LD A,"."
C7C3 12             933   LD (DE),A
C7C4                934  FNWRKQ
C7C4 D1             935   POP DE
C7C5 C9             936   RET
C7C6                937
C7C6                938  ;
C7C6                939  FNP
C7C6 C5             940   PUSH BC
C7C7 D5             941   PUSH DE
C7C8 E5             942   PUSH HL
C7C9 3A 68 C2       943   LD A,(DU)
C7CC 4F             944   LD C,A
C7CD 1A             945   LD A,(DE) ;CURSOR POS.
C7CE 91             946   SUB C
C7CF 6F             947   LD L,A
C7D0 3A 75 C2       948   LD A,(USP)
C7D3 85             949   ADD A,L
C7D4 6F             950   LD L,A
C7D5 3A 76 C2       951   LD A,(YY)
C7D8 85             952   ADD A,L
C7D9 6F             953   LD L,A
C7DA 26 00          954   LD H,0 ;YY+N-DU+USP
C7DC 13             955   INC DE
C7DD 1A             956   LD A,(DE)
C7DE 32 71 C2       957   LD (WWWF+1),A
C7E1 3A 6E C2       958   LD A,(XX)
C7E4 5F             959   LD E,A
C7E5 16 00          960   LD D,0        ;XX
C7E7 CD 63 CA       961   CALL XYADR    ;BC=ADR
C7EA CD 84 CA       962   CALL TURBO?
C7ED DD 21 B2 CA    963   LD IX,WKPRT
C7F1 3E 28          964   LD A,40       ;MOJISU
C7F3 DD 77 00       965   LD (IX+0),A
C7F6 CD 10 C8       966   CALL ATR2
C7F9 21 40 C2       967   LD HL,WORK
C7FC                968  ACKLP1
C7FC DD 70 02       969      LD (IX+2),B
C7FF CD 55 C8       970      CALL ACKPRT
C802 DD 46 02       971      LD B,(IX+2)
C805 23             972      INC HL
C806 03             973      INC BC
C807 DD 35 00       974      DEC (IX+0)
C80A 20 F0          975      JR NZ,ACKLP1
C80C E1             976   POP HL
C80D D1             977   POP DE
C80E C1             978   POP BC
C80F C9             979   RET
C810                980
C810                981  ATR2
C810 AF             982   XOR A
C811 DD 77 03       983   LD (IX+3),A
C814 DD 77 04       984   LD (IX+4),A
C817 3A 71 C2       985   LD A,(WWWF+1)
C81A F6 07          986   OR 7 ;WHITE
C81C DD 77 01       987   LD (IX+1),A
C81F E6 08          988   AND 8
C821 C0             989   RET NZ
C822 3A 73 C2       990   LD A,(WWWF+3)
C825 FE 04          991   CP 4
C827 3E 06          992   LD A,6 ;YELLOW
C829 CA 51 C8       993   JP Z,ATRND
C82C 3A 74 C2       994   LD A,(WWWF+4)
C82F B7             995   OR A
C830 3E 03          996   LD A,3 ;MAZENDA
C832 C2 51 C8       997   JP NZ,ATRND
C835                998
C835 3A 72 C2       999   LD A,(WWWF+2)
C838 B7            1000   OR A
C839 28 05         1001   JR Z,ATRNKK
C83B 16 04         1002     LD D,4
C83D DD 72 03      1003     LD (IX+3),D
C840               1004  ATRNKK
C840 3A 73 C2      1005   LD A,(WWWF+3)
C843 B7            1006   OR A
C844 C8            1007   RET Z
C845 1E 05         1008   LD E,5
C847 FE 03         1009   CP 3
C849 28 02         1010   JR Z,ATRIXS
C84B 1E 04         1011     LD E,4
C84D               1012  ATRIXS
C84D DD 73 04      1013   LD (IX+4),E
C850 C9            1014   RET
C851               1015
C851               1016  ATRND
C851 DD 77 01      1017   LD (IX+1),A
C854 C9            1018   RET
C855               1019
```


▶１年くらい前から仕事でＣ言語を使っていますが，覚えたてのころのソースを見てみると結構笑えるようなことをしていて面白いです。みなさんもたまには昔のソースなど見てみると面白いかもしれませんよ！

宮下　敦(20)神奈川県

```
C855             1020  ;IX+0 +1      +3+4
C855             1021  ;  40 WWWF+1 OF.ADR
C855             1022  ACKPRT
C855 3A B1 CA    1023   LD A,(TURBO)
C858 B7          1024   OR A
C859 28 10       1025   JR Z,NOKANJI
C85B F3          1026   DI
C85C 3E 1D       1027   LD A,$1D
C85E D3 00       1028   OUT (0),A
C860 7E          1029   LD A,(HL)
C861 CD 99 30    1030   CALL SFTCHK@
C864 30 29       1031   JR NC,OKKANJI
C866 3E 1E       1032   LD A,$1E
C868 D3 00       1033   OUT (0),A
C86A FB          1034   EI
C86B             1035  NOKANJI
C86B DD 7E 00    1036   LD A,(IX+0)
C86E DD 5E 01    1037   LD E,(IX+1)
C871 DD 56 03    1038   LD D,(IX+3)
C874 FE 13       1039   CP 19
C876 CC D2 C8    1040   CALL Z,NKK
C879 FE 12       1041   CP 18
C87B CC D2 C8    1042   CALL Z,NKK
C87E FE 11       1043   CP 17
C880 CC D2 C8    1044   CALL Z,NKK
C883 FE 08       1045   CP 8
C885 CC D8 C8    1046   CALL Z,NKK2
C888 7B          1047   LD A,E
C889 5E          1048   LD E,(HL)
C88A 16 00       1049   LD D,0
C88C C3 B5 C8    1050   JP KANJISB
C88F             1051
C88F             1052  OKKANJI
C88F 56          1053   LD D,(HL)
C890 23          1054   INC HL
C891 5E          1055   LD E,(HL)
C892 2B          1056   DEC HL
C893 CD 81 2F    1057   CALL SFTJIS@
C896 CD B6 2F    1058   CALL JISVRM@ ;A,D,E
C899 DD B6 01    1059   OR (IX+1)
C89C CD B5 C8    1060   CALL KANJISB
C89F DD 46 02    1061   LD B,(IX+2)
C8A2 03          1062   INC BC
C8A3 DD 70 02    1063   LD (IX+2),B
C8A6 23          1064   INC HL
C8A7 DD 35 00    1065   DEC (IX+0)
C8AA CB F2       1066   SET 6,D
C8AC CD B5 C8    1067   CALL KANJISB
C8AF 3E 1E       1068   LD A,$1E
C8B1 D3 00       1069   OUT (0),A
C8B3 FB          1070   EI
C8B4 C9          1071   RET
C8B5             1072
C8B5             1073  KANJISB
C8B5 F5          1074   PUSH AF
C8B6 F5          1075   PUSH AF
C8B7 DD 7E 02    1076   LD A,(IX+2)
C8BA C6 20       1077   ADD A,$20
C8BC 47          1078   LD B,A
C8BD F1          1079   POP AF
C8BE ED 79       1080   OUT (C),A
C8C0             1081
C8C0 DD 7E 02    1082   LD A,(IX+2)
C8C3 C6 38       1083   ADD A,$38
C8C5 47          1084   LD B,A
C8C6 ED 51       1085   OUT (C),D
C8C8             1086
C8C8 DD 7E 02    1087   LD A,(IX+2)
C8CB C6 30       1088   ADD A,$30
C8CD 47          1089   LD B,A
C8CE ED 59       1090   OUT (C),E
C8D0 F1          1091   POP AF
C8D1 C9          1092   RET
C8D2             1093
C8D2             1094   ;X1では
C8D2             1095   ;$F8D6 まで
C8D2             1096   ;NOP
C8D2             1097
C8D2             1098  NKK
C8D2 7A          1099   LD A,D
C8D3 B7          1100   OR A
C8D4 C8          1101   RET Z
C8D5 5A          1102   LD E,D
C8D6 AF          1103   XOR A
C8D7 C9          1104   RET
C8D8             1105  NKK2
C8D8 DD 56 04    1106   LD D,(IX+4)
C8DB 7A          1107   LD A,D
C8DC B7          1108   OR A
C8DD C8          1109   RET Z
C8DE 5A          1110   LD E,D
C8DF AF          1111   XOR A
C8E0 C9          1112   RET
C8E1             1113
C8E1             1114  ;
C8E1             1115  FNMAKE
C8E1 C5          1116   PUSH BC
C8E2 D5          1117   PUSH DE
C8E3 E5          1118   PUSH HL
C8E4 ED 53 BE CA 1119   LD (WKDEA),DE
C8E8 21 40 C2    1120   LD HL,WORK
C8EB 11 41 C2    1121   LD DE,WORK+1
C8EE 01 27 00    1122   LD BC,39
C8F1 3E 20       1123   LD A," "
C8F3 77          1124   LD (HL),A
C8F4 ED B0       1125   LDIR
C8F6 ED 5B BE CA 1126   LD DE,(WKDEA)
C8FA 1A          1127   LD A,(DE)
C8FB B7          1128   OR A
C8FC 20 06       1129   JR NZ,TRSSK0
C8FE CD C0 C9    1130     CALL OYADIR
C901 C3 3D C9    1131     JP TRSQ
C904             1132  TRSSK0
C904 FE 01       1133   CP 1
C906 20 09       1134   JR NZ,TRSSK1
C908 2A 78 C2    1135     LD HL,(NWADR)
C90B 22 C0 CA    1136     LD (WTOPA),HL
C90E C3 2F C9    1137     JP LPEND
C911             1138  TRSSK1
C911 5F          1139   LD E,A ;NEW
C912 2A C0 CA    1140   LD HL,(WTOPA)
C915 3A 6A C2    1141   LD A,(CB)
C918 FE 01       1142   CP 1
C91A 20 06       1143   JR NZ,NWNEW
C91C 2A 78 C2    1144     LD HL,(NWADR)
C91F 22 C0 CA    1145     LD (WTOPA),HL
C922             1146  NWNEW
C922 93          1147   SUB E
C923 28 0A       1148   JR Z,LPEND
C925 57          1149   LD D,A ;OLD-NEW
C926 DC CF C9    1150   CALL C,DWNSK
C929 D4 E5 C9    1151   CALL NC,UPSK
C92C 22 C0 CA    1152   LD (WTOPA),HL
C92F             1153  LPEND
C92F 11 47 C2    1154   LD DE,WORK+7
C932 CD FE C9    1155   CALL SBTRS
C935 ED 5B BE CA 1156   LD DE,(WKDEA)
C939 1A          1157   LD A,(DE)
```

```
C93A 32 6A C2    1158   LD (CB),A
C93D             1159  TRSQ
C93D CD 4F C9    1160   CALL MAKEMK
C940 CD 09 CA    1161   CALL ATRSET
C943 ED 5B BE CA 1162   LD DE,(WKDEA)
C947 1A          1163   LD A,(DE)
C948 32 7C C2    1164   LD (CB2),A
C94B E1          1165   POP HL
C94C D1          1166   POP DE
C94D C1          1167   POP BC
C94E C9          1168   RET
C94F             1169
C94F             1170  MAKEMK
C94F AF          1171   XOR A
C950 32 74 C2    1172   LD (WWWF+4),A
C953 3A 60 C2    1173   LD A,(WORK+32)
C956 FE 44       1174   CP "D"
C958 21 C8 CA    1175   LD HL,MRKDR
C95B 28 15       1176   JR Z,MAKMKSK
C95D CD 7B C9    1177     CALL DIRIDF
C960 D8          1178     RET C
C961 ED 5B BE CA 1179     LD DE,(WKDEA)
C965 CD A3 C9    1180     CALL RRCA ;A
C968 A6          1181     AND (HL)
C969 C8          1182     RET Z
C96A 21 CD CA    1183     LD HL,MRKMK
C96D 3E 01       1184     LD A,1
C96F 32 74 C2    1185     LD (WWWF+4),A
C972             1186  MAKMKSK
C972 11 41 C2    1187   LD DE,WORK+1
C975 01 05 00    1188   LD BC,5
C978 ED B0       1189   LDIR
C97A C9          1190   RET
C97B             1191
C97B             1192  DIRIDF
C97B 21 58 CD    1193   LD HL,NMDIR1
C97E 11 D8 CD    1194   LD DE,NMDIR2
C981 06 80       1195   LD B,128
C983             1196  DIRIDFLP
C983 1A          1197     LD A,(DE)
C984 BE          1198     CP (HL)
C985 20 06       1199     JR NZ,NOIDFD
C987 13          1200     INC DE
C988 23          1201     INC HL
C989 10 F8       1202     DJNZ DIRIDFLP
C98B B7          1203     OR A
C98C C9          1204     RET
C98D             1205  NOIDFD
C98D 78          1206   LD A,B
C98E FE 04       1207   CP 4
C990 DC 95 C9    1208   CALL C,MRKCLR
C993 37          1209   SCF
C994 C9          1210   RET
C995             1211
C995             1212  MRKCLR
C995 11 49 CD    1213   LD DE,MRKSTK+1
C998 21 48 CD    1214   LD HL,MRKSTK
C99B 01 0F 00    1215   LD BC,15
C99E AF          1216   XOR A
C99F 77          1217   LD (HL),A
C9A0 ED B0       1218   LDIR
C9A2 C9          1219   RET
C9A3             1220
C9A3             1221  RRCA ;DE
C9A3 1A          1222   LD A,(DE)
C9A4 CB 3F       1223   SRL A
C9A6 CB 3F       1224   SRL A
C9A8 CB 3F       1225   SRL A
C9AA 4F          1226   LD C,A
C9AB 06 00       1227   LD B,0
C9AD 21 48 CD    1228   LD HL,MRKSTK
C9B0 09          1229   ADD HL,BC
C9B1 1A          1230   LD A,(DE)
C9B2 E6 07       1231   AND 7
C9B4 0E 80       1232   LD C,$80
C9B6             1233  RRCALP
C9B6 B7          1234     OR A
C9B7 28 05       1235     JR Z,RRCASK
C9B9 CB 09       1236     RRC C
C9BB 3D          1237     DEC A
C9BC 18 F8       1238     JR RRCALP
C9BE             1239  RRCASK
C9BE 79          1240   LD A,C
C9BF C9          1241   RET
C9C0             1242
C9C0             1243  OYADIR
C9C0 11 47 C2    1244   LD DE,WORK+7
C9C3 3E 2E       1245   LD A,"."
C9C5 12          1246   LD (DE),A
C9C6 13          1247   INC DE
C9C7 12          1248   LD (DE),A
C9C8 11 60 C2    1249   LD DE,WORK+32
C9CB 3E 44       1250   LD A,"D"
C9CD 12          1251   LD (DE),A
C9CE C9          1252   RET
C9CF             1253
C9CF             1254  DWNSK
C9CF F5          1255   PUSH AF
C9D0 7A          1256   LD A,D
C9D1 ED 44       1257   NEG
C9D3 57          1258   LD D,A
C9D4 01 17 00    1259   LD BC,23
C9D7             1260  DWNSLP
C9D7 09          1261     ADD HL,BC
C9D8             1262  FINDCR
C9D8 CD 12 E0    1263       CALL #LDAHL
C9DB 23          1264       INC HL
C9DC FE 0D       1265       CP $0D
C9DE 20 F8       1266       JR NZ,FINDCR
C9E0 15          1267     DEC D
C9E1 20 F4       1268     JR NZ,DWNSLP
C9E3 F1          1269   POP AF
C9E4 C9          1270   RET
C9E5             1271
C9E5             1272  UPSK
C9E5 F5          1273   PUSH AF
C9E6 01 19 00    1274   LD BC,25
C9E9             1275  UPSLP
C9E9 B7          1276     OR A
C9EA ED 42       1277     SBC HL,BC
C9EC CD F5 C9    1278     CALL FINDCRB
C9EF 23          1279     INC HL
C9F0 15          1280     DEC D
C9F1 20 F6       1281     JR NZ,UPSLP
C9F3 F1          1282   POP AF
C9F4 C9          1283   RET
C9F5             1284
C9F5             1285  FINDCRB
C9F5 CD 12 E0    1286   CALL #LDAHL
C9F8 FE 0D       1287   CP $0D
C9FA C8          1288   RET Z
C9FB 2B          1289   DEC HL
C9FC 18 F7       1290   JR FINDCRB
C9FE             1291
C9FE             1292  SBTRS
C9FE CD 12 E0    1293   CALL #LDAHL
CA01 FE 0D       1294   CP $0D
CA03 C8          1295   RET Z
CA04 23          1296   INC HL
```

```
CA05 12          1297   LD (DE),A
CA06 13          1298   INC DE
CA07 18 F5       1299   JR SBTRS
CA09             1300
CA09             1301  ;ATRIBUTE SET
CA09             1302  ATRSET
CA09 AF          1303   XOR A
CA0A 32 72 C2    1304   LD (WWWF+2),A
CA0D 32 73 C2    1305   LD (WWWF+3),A
CA10 3A 60 C2    1306   LD A,(WORK+32)
CA13 FE 44       1307   CP "D"
CA15 20 06       1308   JR NZ,ATRSTSK
CA17 3E 04       1309     LD A,4 ;DIR
CA19 32 73 C2    1310     LD (WWWF+3),A
CA1C C9          1311     RET
CA1D             1312  ATRSTSK
CA1D 21 C2 CA    1313   LD HL,KKDM
CA20 CD 42 CA    1314   CALL KKXB
CA23 21 C5 CA    1315   LD HL,KKDM+3
CA26 CD 42 CA    1316   CALL KKXB
CA29 16 03       1317   LD D,3
CA2B 3A 60 C2    1318   LD A,(WORK+32)
CA2E FE 4D       1319   CP "M"
CA30 28 0B       1320   JR Z,ATRSTSK2
CA32 16 01       1321   LD D,1
CA34 FE 42       1322   CP "B"
CA36 28 05       1323   JR Z,ATRSTSK2
CA38 16 02       1324   LD D,2
CA3A FE 41       1325   CP "A"
CA3C C0          1326   RET NZ
CA3D             1327  ATRSTSK2
CA3D 7A          1328   LD A,D
CA3E 32 73 C2    1329   LD (WWWF+3),A
CA41 C9          1330   RET
CA42             1331
CA42             1332  KKXB
CA42 C5          1333   PUSH BC
CA43 11 55 C2    1334   LD DE,WORK+21
CA46 06 03       1335   LD B,3
CA48             1336  KKXBLP
CA48 1A          1337     LD A,(DE)
CA49 13          1338     INC DE
CA4A CD 5A CA    1339     CALL OKCHG
CA4D BE          1340     CP (HL)
CA4E 23          1341     INC HL
CA4F 20 07       1342     JR NZ,KKXBSK
CA51 10 F5       1343     DJNZ KKXBLP
CA53 3E 01       1344   LD A,1
CA55 32 72 C2    1345   LD (WWWF+2),A
CA58             1346  KKXBSK
CA58 C1          1347   POP BC
CA59 C9          1348   RET
CA5A             1349
CA5A             1350  OKCHG
CA5A FE 61       1351   CP "a"
CA5C D8          1352   RET C
CA5D FE 7B       1353   CP "z"+1
CA5F D0          1354   RET NC
CA60 D6 20       1355   SUB "a"-"A"
CA62 C9          1356   RET
CA63             1357
CA63             1358  ;
CA63             1359  XYADR  ;DE,HL=X,Y BC=ADR
CA63 3A 6F C2    1360   LD A,(WIDTH)
CA66 FE 28       1361   CP 40
CA68 CA 78 CA    1362   JP Z,L40
CA6B             1363
CA6B 29          1364   ADD HL,HL
CA6C 29          1365   ADD HL,HL
CA6D 29          1366   ADD HL,HL
CA6E 29          1367   ADD HL,HL
CA6F E5          1368   PUSH HL    ;Y*16
CA70 29          1369   ADD HL,HL
CA71 29          1370   ADD HL,HL ;Y*64
CA72 19          1371   ADD HL,DE ;Y*64+X
CA73 D1          1372   POP DE
CA74 19          1373   ADD HL,DE ;Y*64+X+Y*16
CA75 44          1374   LD B,H
CA76 4D          1375   LD C,L
CA77 C9          1376   RET
CA78             1377
CA78             1378  L40
CA78 29          1379   ADD HL,HL
CA79 29          1380   ADD HL,HL
CA7A 29          1381   ADD HL,HL
CA7B E5          1382   PUSH HL    ;Y*8
CA7C 29          1383   ADD HL,HL
CA7D 29          1384   ADD HL,HL
CA7E 19          1385   ADD HL,DE ;Y*32+X
CA7F D1          1386   POP DE
CA80 19          1387   ADD HL,DE ;Y*32+X+Y*8
CA81 44          1388   LD B,H
CA82 4D          1389   LD C,L
CA83 C9          1390   RET
CA84             1391
CA84             1392  ;
CA84             1393  TURBO?
CA84 C5          1394   PUSH BC
CA85 3A 77 C2    1395   LD A,(KMODE)
CA88 B7          1396   OR A
CA89 3E 00       1397   LD A,0
CA8B 20 1F       1398   JR NZ,TRB
CA8D 3A 7F D0    1399   LD A,($D07F)
CA90 B7          1400   OR A
CA91 28 19       1401   JR Z,TRB
CA93 F3          1402   DI
CA94 3E 1D       1403   LD A,$1D
CA96 ED 79       1404   OUT (C),A
CA98 3A 00 00    1405   LD A,(0)
CA9B 4F          1406   LD C,A
CA9C 3A 00 10    1407   LD A,($1000)
CA9F 47          1408   LD B,A
CAA0 3E 1E       1409   LD A,$1E
CAA2 D3 00       1410   OUT (0),A
CAA4 FB          1411   EI
CAA5 78          1412   LD A,B
CAA6 B9          1413   CP C
CAA7 3E 01       1414   LD A,1
CAA9 20 01       1415   JR NZ,TRB
CAAB AF          1416   XOR A
CAAC             1417  TRB
CAAC 32 B1 CA    1418   LD (TURBO),A
CAAF C1          1419   POP BC
CAB0 C9          1420   RET
CAB1             1421
CAB1 00          1422  TURBO DB 0 ;0..X1
CAB2             1423  WKPRT DS 5
CAB7             1424  WXYSS DS 7
CABE 00 00       1425  WKDEA DW 0
CAC0 00 00       1426  WTOPA DW 0
CAC2 58 31 20 42 1427  KKDM  DM "X1 BAT"
CAC6 41 54
CAC8 3C 44 49 52 1428  MRKDR DM "<DIR>"
CACC 3E
CACD 3C 4D 52 4B 1429  MRKMK DM "<MRK>"
CAD1 3E
CAD2             1430
```

▶Oh!Dynaが出たんですね。オー！ ダイアナみたいですね。　山本 壽一郎(28)愛知県

X68000CARD.FNC用カードゲーム

COUPLE

Aoki Michio 青木 実千男

都合により，先月の予告とは違いますが，ひとり遊びCOUPLEをお届けします。隣接した同じ数のカードを取り除いていく，というルールですべてのカードを取り除いてください。画面いっぱいになるとすーっとスクロールするのが気持ちいい？

遊び方

CARD. FNCを使ったトランプゲームです。CARD. FNCを拡張したX-BASICからそのまま打ち込んでください。

RUNしてしばらくすると“Push To Start”と出ます。このときカードをシャッフルしているので，少し待ってからマウスのボタン（左）を押す。するとカードが1枚出てきます。次からNextのあたり（このへんおおざっぱ）を押すと，次のカードが出てきて，横に4枚並ぶと次からは1段下がります。Lineが4を越えるとスクロールしていきます。

で，これでいったいなにをしろというかというと，COUPLEというひとりで遊ぶトランプゲームなんですよ。

肝心要のルールですが，こうして並んだカードが縦，横，斜めに同じ数字なら取っぱらっちゃって，空いたところは詰めていくわけです。そんで，カードを全部使いきって，全部取っぱらっちゃえば“CONGRATULATION”というわけやね（僕は大阪から越してきてまだ半年なんです。埼玉県加須市在住のX68000ユーザーっていませんか？）。

＊

取っぱらい方は，（左ボタンで）1枚目クリック。2枚目クリック。ここでもう1回クリックするとOKです。キャンセルは右ボタンね。本当の遊び方との違いは，常に最後の4列しか取れないというところです。まあ，遊んでみればわかるでしょう。

あと，Lineは全部で何行あるか，Lastは残り何枚かを表しています。コツは最初4列くらいばばっと出して，取っては出し，取っては出しを繰り返す，これだと思うんだけど……。

プログラムについて

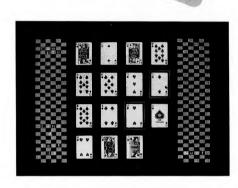
プログラムの概略は表1を見てもらえばわかるでしょう。プログラムで使用している変数は表2にまとめておきました。それから，サウンドの部分とシャッフルルーチンは1990年3月号の“99”を参考にしています。

このゲームは夏休みにちょこちょこ作りました。それから，X-BASICはあんまりいじったことがなかったので苦労したよー。あーあ勉強しなきゃ。いま徹夜明けなので文がしっちゃかめっちゃかだけど許してくださいませ。

表1　内容の説明

行	内容
90～ 150	グローバル変数の宣言
170～ 230	これがメインです
250～ 310	いろんなものの初期設定
320～ 390	画面の両端にタイル模様を描きます
400	そのあとで，ウィンドウエリアを設定し直す
410～ 450	card（ ）に順番にカード番号を入れつつ，date（ ）に変数iの全角文字を入れていく
460～ 510	文字を表示
560～ 590	変数その他の初期化
600～ 660	カードのシャッフル
730～ 760	マウスボタンが押されるのを待つ
770	Nextを押したとき
780	カードをクリックしたとき
790	キャンセルしたとき
860	カードを出す位置を求める
870	それが4列目以降，右端のカードならスクロール
1040	上下どちらにスクロールするかによって，出てくるカードの位置を求める
1050～1070	スクロールしたあとに出てくるカードを描く
1080～1100	スクロール
1160～1190	1回目のクリックの処理
1200～1230	2回目のクリックの処理
1240～1340	3回目のクリックの処理 カードを取って詰めます
1330	4列目以降，左から2枚以内にカードがあったなら逆スクロール
1380～1420	キャンセルしたときの処理
1460～1480	カーソル表示
1520～1530	ウェイト処理
1570～1650	2回目のクリックを認めるのかチェック
1580	そこにカードがあるか
1590	数は同じか
1600	1回目のクリックと同じカードか
1610～1630	縦，横，斜めのいずれかに位置するか
1720	クリアしたなら……
1730～1830	ゲームオーバーの処理
1860～1890	クリアしたときの文字の表示
1900～1930	効果音

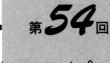
表2　変数表

グローバル変数

変数	内容
card( )	カードが出てくる順番にカード番号が入っている
scrn( )	画面に出てきたカードの位置を記憶しておく
pows( )	クリアしたとき，文字拡大の倍率
date( )	数字の全角文字が入る
ch	“CONGRATULATION”，クリアしたときに使う
pan	クリアしたときに使う音
pon	カードが出たとき，または動いたときに出る音
hy	homeのY座標の位置
frg	何回目のクリックか
hip	次にカードが出る位置
dm	マウス処理のダミー
mx	マウスのX座標
my	マウスのY座標
bl	マウスの左ボタン
br	マウスの右ボタン
pol	1回目にクリックしたカードの位置
kazu1	そのカードの数
po2	2回目にクリックしたカードの位置
kazu2	そのカードの数
num	何番目のカードか

リスト1

```
  10 /* ============================================
  20 /* ===== 至高の CARD・FNC 用ゲーム ======
  30 /* =====                          ======
  40 /* =====          C o u p l e      ======
  50 /* =====                          ======
  60 /* =====          青木実千男       ======
  70 /* ============================================
  80 /*
  90 dim int card(51),scrn(55),pows(4)={0,1,6,17,36}
 100 dim str data(51)
 110 str ch="CONGRATULATION",pan="@68o3v1514c",pon="@59o3v1514c
"
 120 int hy,frg,hip
 130 int dm,mx,my,bl,br
 140 int po1,kazu1,po2,kazu2
 150 int num
 160 /* ============================================
 170 init()
 180 while 1
 190   cinit()
 200   game()
 210   gameover()
 220 endwhile
 230 end
 240 /* ---------   初期設定
 250 func init()
 260   int x,y,c=0,i
 270   screen 1,0,1,1
 280   console 0,32,0
 290   mouse(0):mouse(4):mouse(1)
 300   window(0,0,1023,1023)
 310   m_alloc(1,200):m_assign(1,1)
 320   for y=0 to 63
 330     c=(y mod 2)*14
 340     for x=0 to 5
 350     if c=14 then c=0 else c=14
 360     fill(x*16,y*16,x*16+15,y*16+15,c)
 370     fill(x*16+415,y*16,x*16+415+15,y*16+15,c)
 380     next
 390   next
 400   window(112,0,398,1023)
 410   for i=0 to 51
 420     card(i)=i+1
 430     if i<10 then { data(i)=" "+chr$(130)+chr$(i+79)+" "
 440       } else data(i)=chr$(130)+chr$(val(left$(str$(i),1))+
79)+chr$(130)+chr$(val(right$(str$(i),1))+79)
 450   next
 460   color 7
 470   locate  2, 2:print"C o u";
 480   locate  2, 3:print"  p l e";
 490   locate  2,22:print"L i n e";
 500   locate  2,27:print"L a s t";
 510   locate 54,29:print"N e x t";
 520 endfunc
 530 /* ---------   カードの初期化
 540 func cinit()
 550   int a,b,c,i
 560   wipe():home(0,0,0)
 570   for i=0 to 54:scrn(i)=54:next
 580   hip=-1:hy=0:frg=0:num=0
 590   score()
 600   locate 19,10:print"P u s h   T o   S t a r t";
 610   repeat
 620     a=rnd()*52:b=rnd()*52
 630     c=card(a):card(a)=card(b):card(b)=c
 640     msstat(dm,dm,bl,dm)
 650   until bl=-1
 660   locate 19,10:print spc(26);
 670 endfunc
 680 /* ---------   ゲーム
 690 func game()
 700   for num=0 to 51
 710     shori()
 720     while 1
 730       repeat
 740         msstat(dm,dm,bl,br)
 750       until bl=-1 or br=-1
 760       mspos(mx,my)
 770       if mx>416 and my>432 and frg=0 then wait(200):break
 780       if mx>111 and mx<400 and bl=-1 then toru()
 790       if br=-1 then cancel()
 800       wait(700)
 810     endwhile
 820   next
 830 endfunc
 840 /* ---------   カードを出す
 850 func shori()
 860   hip=hip+1
 870   if hip>15 and (hip mod 4)=0 then scroll(+1)
 880   card_set(hip,card(num))
 890   score()
 900 endfunc
 910 /* ---------   インフォメーション
 920 func score()
 930   locate 4,24:print data(hip/4+1);
 940   locate 4,29:print data(51-num);
 950 endfunc
 960 /* ---------   カード表示
 970 func card_set(po,n)
 980   oto(pon):scrn(po)=n
 990   c_put((po mod 4)*72+124,(po/4*128+16) mod 1024,n)
1000 endfunc
1010 /* ---------   スクロール
1020 func scroll(d;int)
1030   int i,sy
1040   if d=1 then sy=hip/4+1 else sy=hip/4-4
1050   for i=0 to 3
1060     c_put(i*72+124,(sy*128+16) mod 1024,scrn(i+sy*4))
1070   next
1080   for i=1 to 128
1090     hy=hy+d:home(0,0,hy mod 1024)
1100   next
1110 endfunc
1120 /* ---------   カード指定
1130 func toru()
1140   int st,ch,d=1,i
1150     switch frg
1160       case 0:po1=(mx-112)/72+(my/128+hy/128)*4
1170              kazu1=scrn(po1) mod 13
1180              if scrn(po1)<54 then set(po1,11):frg=1
1190              break
1200       case 1:po2=(mx-112)/72+(my/128+hy/128)*4
1210              kazu2=scrn(po2) mod 13
1220              if setcheck()=1 then set(po2,11):frg=2
1230              break
1240       case 2:set(po1,0):set(po2,0)
1250              frg=0
1260              if po1<po2 then st=po1:ch=po2-1 else st=po2:c
h=po1-1
1270              for i=st to hip-2
1280                if i=ch then d=2
1290                card_set(i,scrn(i+d))
1300              next
1310              card_set(hip-1,54):card_set(hip,54)
1320              if hip>15 and (hip mod 4)<2 then scroll(-1)
1330              hip=hip-2
1340              score()
1350     endswitch
1360 endfunc
1370 /* ---------   キャンセル
1380 func cancel()
1390   switch frg
1400     case 1:set(po1,0):frg=0:break
1410     case 2:set(po2,0):frg=1
1420   endswitch
1430 endfunc
1440 /* ---------   カーソル表示
1450 func set(po,c)
1460   int x,y
1470   x=(po mod 4):y=po/4
1480   box(x*72+118,(y*128+8) mod 1024,x*72+176,(y*128+118) mod
1024,c)
1490 endfunc
1500 /* ---------   ウェイト
1510 func wait(l)
1520   int i
1530   for i=0 to l:next
1540 endfunc
1550 /* ---------   セットのチェック
1560 func setcheck()
1570   int x1,y1,x2,y2
1580   if scrn(po2)<54 then {
1590     if kazu1=kazu2 then {
1600       if not(po1=po2) then {
1610         x1=po1 mod 4:y1=po1/4
1620         x2=po2 mod 4:y2=po2/4
1630         if x2>x1-2 and x2<x1+2 and y2>y1-2 and y2<y1+2 the
n return(1)
1640         } } }
1650   return(0)
1660 endfunc
1670 /* ---------   ゲームオーバー処理
1680 func gameover()
1690   int i,q
1700   str s
1710   wipe():home(0,0,0)
1720   if scrn(0)=54 then s=ch else s="    REGRET"
1730   symbol(144,632,s,2,1,1,15,0)
1740   for i=1 to 384:home(0,0,i):wait(0):next
1750   if scrn(0)=54 then {
1760     for i=0 to 13
1770       oto(pan)
1780       for q=1 to 4:moji(i,q,15):moji(i,q,0):next
1790       for q=1 to 3:moji(i,5-q,15):moji(i,5-q,0):next
1800       moji(i,1,15)
1810     next
1820     }
1830   for i=385 to 896:home(0,0,i):wait(0):next
1840 endfunc
1850 /* ---------   ゲームオーバーの文字表示
1860 func moji(x,y,c)
1870   y=pows(y)
1880   symbol(144+16*x,640-y*8,mid$(ch,x+1,1),2,y,1,c,0)
1890 endfunc
1900 /* ---------   効果音
1910 func oto(m;str)
1920   m_init():m_trk(1,m):m_play()
1930 endfunc
```

第54回

猫とコンピュータ

マニアの砦にて

Takazawa Kyoko
高沢 恭子

10月14日に秋葉原で行われた「第7回ホビーマイコンショウ」。キョウコさんも久しぶりに会った通信仲間と，楽しく1日を過ごしたようです。和気あいあいのこの様子を，2回にわたってレポートします。

まだ夏の暑さが見えかくれしている10月はじめの昼さがり。

近くの文具屋さんでケント紙を買いこんで，自転車のカゴにポンと入れたとき，そうだ，プリンタ用紙を買っておくように夫に言われたのだったと思い出した。15インチのほうだったかな，10インチだったかなと考えながら，地下鉄東西線N駅のアーケード街に沿った道にまわると，ペンキ屋さんのしごとに出会った。

改装中の店舗の外壁を塗装しているところで，飛沫をさけるための大きなビニールを広い範囲にめぐらせた中で，脚立に乗った若い人が作業をしている。

よく見ると，壁一面に銃で撃ちつけたような塗料のツブツブがあり，その上からハンドローラーをかけて，表面をたいらに整えている。

ああ，不揃いの碁石にプレスをかけたような壁は，こうやってこしらえるのかと，自転車を降りてしばらく見学した。

「塗料を吹き付けてから，どのくらいの時間をあけてローラーをかけるのですか」

とたずねると，

「わりあいすぐやってしまうんですよ。塗料が流れちゃいますからね」

と下で見ていた年長の職人さんが答えてくれた。油絵の具にそっくりの溶剤の匂いが，すこし不透明な日差しの中に溶け込んだようにあたりに広がり，製作中のアイボリー色の壁も，ツナギ服の2人の若い職人も，ほんわり輝いて見える。なんとも心地よい時間だ。

そのとき，となりのブティックから顔見知りの女主人が出てきて，

「アラ，いつもノンキそうでいいわね」

と声をかけた。そうだ，用事が鎖のようにつながっていたんだった。返事がわりのテレ笑いをして，ペンキやさんにお礼をいうと，予定を変えてスーパーに向かった。プリンタ用紙は重量があるから，また出直すことにしよう。

自由のセンタク

「あの，たとえばですけど……ヌイグルミなんかも……クリーニングしてもらえるものなんでしょうか」

スーパーの一角にあるクリーニング屋さんには，いつもの店員さんが2人，カウンターに立っていた。「ノンキでいい」と言われたすぐあとなのでこの質問にはちょっとためらいがあったが，参考に聞いておきたいことだったし店も空いていてよい機会だ。

「扱ってますよ。でもけっこういちばん安くても1,000円はするんですよ」

値段表をまとめた冊子を持ってきて，ヌイグルミのページを開いて見せてくれた。いちばん上のランクは3,000円以上になっているが，けっきょく実物を見ないと決められないようだ。

本物の毛皮，ホンニャアが来てから，それまでトオルのペットだったヌイグルミたち，古参の順に，フル（古）ニャア，シン(新)ニャアと呼ばれていた彼らがどうなったかというと，愛着はじゅうぶんあるものの，長いあいだには転勤などを機に処分されてしまっていた。

ただ1匹，ホンニャアが来る直前に，ある店でみつけた毛足の長い，すこしばかり気どった青い目の猫のヌイグルミだけが，新品のまま押し入れにしまわれていた。ペルシャ猫らしい姿なので，トオルが「ペルニャア」と名づけていた。

ホンニャアが一匹前のオトナになったとき，おもしろ半分にこのペルニャアを彼に見せたところ，いきなり怒ったようにとびつき，ヌイグルミと抱き合うようなかたちで前足で押さえつけ，後足で猛烈なキックの連打をした。相手に生命がないことは本能的にわかると思うのだが，トレーニングのつもりにしては迫真の興奮ぶりで，おそろしいほどだった。それ以後，ペルニャアを決してホンニャアに見せないようにつとめてきた。

それがちかごろ，誰かが捜しものでもしたときにちょっと位置が入れかわったのか，ペルニャアの入ったビニール袋が押し入れの前面にのぞいていた。わが家の押し入れは，通気のために端がいつも10センチほどあいている。私たちはウッカリしていたが，ホンニャアにしてみれば，宿敵との再会だった。

彼はさっそく袋ごとペルニャアをひきずり出して格闘したらしい。発見されたとき，真新しかったペルニャアは，ビニール袋が破られ，玄関のタタキの上で汚れ色になって倒れていた。

「洗濯機で洗っちゃったらどうですか」

もうひとりの店員さんが言った。

「私，孫のヌイグルミを洗濯機で洗いましたよ。そのあと3日間，外に干しっぱなしにして乾かしましたけどね」

スゴイ！　なんという迷いのない力強いやりかた。汚れを落とすならそれがいちばんだ。私はといえば，あのフワフワの風合いを失うまいと，ぬるま湯，毛糸洗剤，手洗い，最短時間，陰干しという臆病な条件で，いままで何回も挑戦しながら，いつも失望をくりかえしてきたのに。だから，こんどのペルニャアは，汚れ落としより風合いを優先にして，ドライクリーニングを考えたのだ。

「それで，仕上がりはどうでしたか？」

「真っ白になりましたよぉ」

お孫さんがいるにしてはまだとても若いその人は，満足そうに言った。洗濯機の中で目をまわし，３昼夜戸外にさらされて，疲れてカラカラになったヌイグルミが目に浮かんできたが，洗濯というのは仕上がりにかける期待によって，こんなに奔放なやりかたがセンタク（選択）できるものなのだと気がついた。

さて，買い物をすませたら，ホビーショウのポスターをつくらなくては。

隠し砦の善人たち

雨の宣告を受けていた10月14日は，朝からだんだん空が明るくなり，日中は汗ばむほどの日となった。

「第７回ホビーマイコンショウ」は，いつものように，秋葉原ラジオ会館８階大ホールで開催された。創立11周年の「きまぐれコンピュータクラブ」と，同じく10周年の「FORESIGHT」(フォーサイト)，それに開局５年目を迎えた「FBI-NET」の共催である。

プロ，アマをまじえたメンバーが，ハード，ソフトのオリジナル作品を披露しながら，親睦と情報交換の１日をすごすのが目的で，誰でも無料で入場できる。

ただでさえ行事の多い季節に，今年は科学技術館で毎年開かれる「全国草の根ＢＢＳ大会」が１週間後と，日程が接近してしまい，スタッフも共通していることから，みんなたいへん忙しい思いをした。

いつも「踊る人形」などの人気作が話題のコバヤシ先生は，予定の「文福茶釜」が故障で，「デジタル気圧計」を出品。金沢の高校の数学の先生，ワカマツさんは，継続テーマである「πの計算」を，前回の15万桁から，100万桁まで算出に成功。FM TOWNSによるCGのデモとあわせて出品された。

イシイさんは，中古基板を組み合わせた自作のゲーム。群馬のタムラさんは，「音声認識の実験」。画面に向かって口笛を吹くと，マイクから入力した音をA/D変換してZ80ボードで処理。PC-6001で同じ音程を再現する。

広島の理科の先生イマオカさんは，教材としての目的で開発した，MSX2による「圧力測定システム」。測定の結果をグラフ表示もする。ソフト制作が専門のナカニシさんも，CAIの例として，「化学の尺度モル」の概念をわかりやすく説明するグラフィックを出品，デモをしてみせた。

実業家のオザキさんは，手作りのガイガーミュラー管による「宇宙線センサー」，広い大気中に飛び交う，目に見えない「宇宙線」を感知して音で知らせる。

夫はショウ開催の段取りに時間を取られていたが，人間が近づくと24の小窓が点滅するイルミネーションのセンサーを，装飾をかねて製作した。

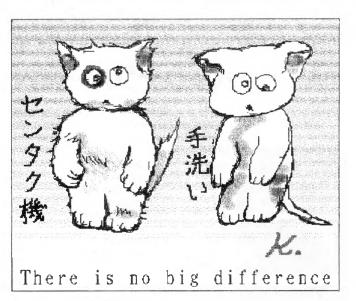
FBI-NETの２大名物は，10回線の端末を使った「パソ通シミュレーション」と，「天麩羅★三杯酢」ことヤマザキさんの「電脳駄菓子屋」さん。

展示のほかに，PA装置完備のトークセッションプログラムとして，BBSの生みの親でもあるヨコタ先生（横田秀次郎氏）の「OS/2 LANマネージャとNetwareの戦い」と題する講演，カワムラ先生（川村清氏）の通信端末プログラムの新モデル「BIG-Term」の発表，ナカニシさんの展示作品「化学の尺度モル」についてのスピーチが行われた。

通信ファミリー

「キョウコさん，エレベータの中のポスターですけどね，猫がもっているGUNはMGCのM75でしょ」

FBIの「ちゃがま」ことナカムラテツヤ君に，また名答されてしまった。ラジオ会館の２基あるエレベータの中に，ホビーショウのポスターとして，１対のコミックを描いた。１枚はICや電子パーツを積んだ要塞から，猫がGUNをかまえているもの。もう１枚は，弾丸を受けて倒れているもので，「マニアの最後の砦」なんてコメントも入れた。

この連載でいつか描いたGUNも，ちゃがま君に「あれはPPKですね」とあっさり当てられて驚いたが，くわしく描きこんだ絵ならともかく，簡単に印象だけで描いたものなのに，いかに熟知しているかということだろう。

FBIのGUNボード「SWAT」とシャレでやっている「きょうこファンクラブ」の両方のシグオペでもあるちゃがま君は，この春から業界大手Ｑ社の社員である。

「みゆ」君，「むーみむ」君，「CLOVIS」君。イベントというと，大きな機動力とタレント性を発揮してくれるFBIの主力メンバーの多くが，この春から社会人となった。

画面の文字を通じて初めて交信する感動をいっしょに体験した仲間たち。半角のカタカナから，全角漢字へ，そしてANSIのカラーグラフィックまで。通信が成長していく歳月と併せて，５年間の貴重な時代を共にすごしたことになる。

そういえば，小学生だった「GETIKOBA」ちゃんはもう高校生，中学生で私にANSIを手ほどきしてくれた「ながみね講師」も来春は大学受験，新聞社のワープロオペレータ「ちやこ」さんは，結婚してママになり，プロ中のプロであるSSKさんも，この間に大手のコンピュータ会社から独立した。

シスオペ「nin隊長」（中村守利氏）の尽力とメンバーの力でネットは熟成し，寂しかった女性ボード「ばななくらぶ」も，一挙に若いメンバーが増えた。

きょうのショウの進行の途中で，その女性ネットワーカーたちを取材させてほしいと，某女性週刊誌から申し込みを受けた。

FBIのメンバー，編集マンの「銀猫」さんを通じての依頼で，約束の３時に記者氏がカメラマンをともなってあらわれた。

（以下次号へ）

バックナンバー案内

BACK ISSUES

ここには1989年12月号から1990年11月号までをご紹介しました。現在1989年10~12, 1990年2, 4, 6, 8~11月号までの在庫がございます。バックナンバーおよび定期講読のお申し込み方法については, 186ページを参照してください。

1989

12月号
特集 Cプログラミングへの招待
付録 C言語簡易リファレンス
連載 ショートプロぱーてぃ/Z80's Bar
X68000マシン語/X-BASIC/DōGA・CGA
●Oh!X2周年特別企画「素粒子の声が聞こえる」
●X1/turbo用アクションゲームACTIVE UNIT
LIVE in '89 天空の城ラピュタ/ギャラクシーフォース
THE SOFTOUCH 38万キロの虚空/た~みのる2
全機種共通システム SLANG用リダイレクションライブラリ

1990

1月号(品切れ)
特集1 オペレーティングスタイルの研究
特集2 Cプログラミング応用編
連載 ショートプロぱーてぃ/Z80's Bar
X68000マシン語/C調言語講座/DōGA・CGA
●X1/turbo用シミュレーションゲームSuper Battle
LIVE in '90 さよならを過ぎて/RYDEEN
THE SOFTOUCH レナム/メタルサイト
全機種共通システム WORM KUN/再掲載SLANG
特別付録 X68000 THE SOFTWARE CATALOGUE

2月号
特集 画像圧縮へのアプローチ
連載 ショートプロぱーてぃ/Z80's Bar/DōGA・CGA
X68000マシン語/C調言語講座/X-BASIC調理実習
●X68000用ゲームプログラムGon Gon
●MZ-700用紙芝居Eyelarth
LIVE in '90 オーダイン/魔女の宅急便
THE SOFTOUCH A-JAX/フラッピー2/夢幻戦士ヴァリスII
マジックパレット/Mu-1/CYBERNOTE PRO-68K
全機種共通システム 超小型コンパイラTTC++

3月号 (品切れ)
特集 MUSICアドベンチャー
X68000用MIDIドライバ&音源エディタ
なんでも鳴らせるOPMD.X/MMLを楽譜データに
連載 ショートプロぱーてぃ/Z80's Bar/DōGA・CGA
C調言語講座/X-BASIC調理実習
●X1/turboシミュレーションCRISIS in Tokyo
LIVE in '90 パワードリフト/スキーム/となりのトトロ
THE SOFTOUCH ナイトアームズ/斬/ダンジョンマスター
全機種共通システム 超多機能アセンブラOHM-Z80

4月号
特集 ゲームシステム文学誌
1989年度GAME OF THE YEAR発表
連載 ショートプロぱーてぃ/Z80's Bar/DōGA・CGA
X-BASIC調理実習/C調言語講座/X68000マシン語
●X1・MZ-2000/2500用RPG The Cave of Dalk
●うわさの68040, ついに登場
LIVE in '90 バーニングフォース(OPMD対応)
THE SOFTOUCH The Fille Professor/HOST PRO-68K
全機種共通システム ファジィコンピュータシミュレータI-MY

5月号(品切れ)
特集 BASICプログラミング
第5回 言わせてくれなくちゃだワ
連載 ショートプロぱーてぃ/Z80's Bar
X-BASIC調理実習/X68000マシン語プログラミング
●新機種X68000SUPER-HD/EXPERTII/PROII
●ラジコンスティックの製作
LIVE in '90 TURBO OUTRUN
THE SOFTOUCH 天下統一/ポピュラス/Hyperword
全機種共通システム インタプリタ言語STACK

6月号
特集 創刊8周年記念PRO-68K(付録5"2HD)
Oh!Xアンケート結果大分析大会
連載 ショートプロぱーてぃ/Z80's Bar/PurePASCAL
X-BASIC調理実習/X68000マシン語プログラミング
●X1turbo用コマンドシェルシミュレータ
●ハードウェア工作入門
LIVE in '90 ナイトアームズ/悪魔城伝説/この木なんの木
THE SOFTOUCH 三国志II/FAR SIDE MOON/グラナダ
全機種共通システム X68000用S-OS"SWORD"他

7月号(品切れ)
特集 マシン語への第一歩
X68000SUPER-HD試用レポート
連載 ショートプロぱーてぃ/Z80's Bar/DōGA・CGA
X-BASIC調理実習/PurePASCAL
●INTEGRAL X1——ノーマルX1への対応
●ハードウェア工作入門
LIVE in '90 夢幻戦士ヴァリスII/トッカータとフーガニ短調
THE SOFTOUCH サーク/あーくしゅ/ダウンタウン熱血物語
全機種共通システム リロケータブルアセンブラWZD

8月号
特集 ADVANCED 2D GRAPHICS
100号記念特別モニタプレゼント
連載 ショートプロぱーてぃ/Z80's Bar/INTEGRAL X1
X-BASIC調理実習/X68000マシン語プログラミング
PurePASCAL/ハードウェア工作入門
●X68000用画像回転プログラム XROT0.X
LIVE in '90 OMENS OF LOVE/ENDLESS RAIN/ダートフォックス
THE SOFTOUCH 大航海時代/ウルティマV/プロミストランド
全機種共通システム リンカWLK

9月号
特集1 日本語を処理するための序章
特集2 ADVANCED 2D GRAPHICS
連載 ショートプロぱーてぃ/Z80's Bar/DōGA・CGA
X-BASIC調理実習/マシン語プログラミング
PurePASCAL/ハードウェア工作入門
●清水和人流プログラミング道場
LIVE in '90 風の谷のナウシカ/ラジオ体操第一
THE SOFTOUCH T&T/D-Again/シムシティー/ギャラガ'88ほか
全機種共通システム BILLIARDS

10月号
特集 電子音楽術入門
連載 ショートプロぱーてぃ/Z80's Bar/DōGA・CGA
マシン語プログラミング/ハードウェア工作入門
清水和人流プログラミング道場
●荻窪圭の大人のためのX68000
●中森章のようこそここへC言語
LIVE in '90 Rise And Fall/PARADOX/キューピー3分クッキング
THE SOFTOUCH ワールドコート/ルーンワース/闇の血族/提督の決断
全機種共通システム ライブラリアンWLB

11月号
特集 理科系のGAME REVIEW
連載 Z80's Bar/DōGA・CGA/カードゲーム
マシン語プログラミング/ハードウェア工作入門
PurePASCAL/X-BASIC調理実習
ようこそここへC言語/INTEGRAL X1
荻窪圭の大人のためのX68000
LIVE in '90 ピラミッドソーサリアン/ザ・スキーム
THE SOFTOUCH SPECIAL ラグーン/幻獣鬼/サイバリオン/GUNSHIP他
全機種共通システム スクリーンエディタEDC-T

●編集部へのメッセージ

今月号の特集について

いちばん良かった記事	興味のなかった記事
これから載せてほしい記事内容	本誌以外にお読みのパソコン雑誌

推薦する市販ソフト

ソフト名：

推薦理由：

あなたのパソコンの拡張スロットには何がささっていますか？

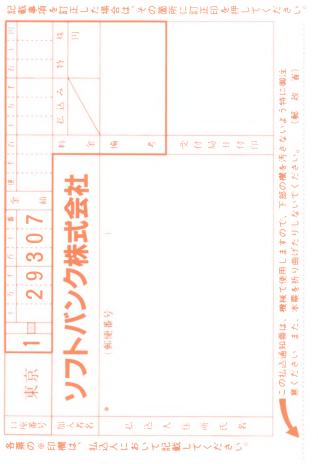
あなたの愛機は(所有機種に○印をつけてください)　　　ない

X1(マニアタイプ,C,D,F,G,twin) X1 turbo(model 10,20,30,40,II,III,Z,ZII,ZIII)

MZ-(80K/C, 1200, 700, 1500, 80B, 2000, 2200, 2500, 2861)

X68000(初代, ACE,PRO,PRO II,EXPERT,EXPERT II,SUPER, HD)　その他

FD(　基)　TAPE　QD　HD(　MB)　プリンタ(　)

年齢　歳　パソコン歴　年　男・女　プレゼントNo.

各票の※印欄は、払込人において記載してください。

通常払込料金
加入者負担

払込通知票

口座番号	東京 1－29307	金額	億 千 百 十 万 千 百 十 円
加入者名	ソフトバンク株式会社	料金	払込み / 特殊 円
払込人住所氏名	※（郵便番号　　　）	備考	
		受付局日付印	

この払込通知票は、機械で使用しますので、下部の欄を汚さないよう特に御注意ください　また、本票を折り曲げたりしないでください。　（郵政省）

記載事項を訂正した場合は、その箇所に訂正印を押してください。

切り取らないで郵便局にお出しください。

通常払込料金
加入者負担

払込票

口座番号	東京 1－29307
加入者名	ソフトバンク株式会社
金額	億 千 百 十 万 千 百 十 円
払込人住所氏名	※
備考	受付局日付印

振替用紙

←点線から、きれいに切り取ってご使用ねがいます。

切り取り線

定期購読のお申し込みを頂きありがとうございます。この申込書の弊社到着〆切は次の通りです。これを過ぎますと次号からの発送となりますので、ご了承下さい。尚、郵便振替は郵便局で払い込まれてから当社に到着するのに、2週間位かかります。

Oh!PC	1日発売	前月 20日
	15日発売	当月 5日
月刊情報処理試験 BEEP メガドライブ		前月 末日
Oh! X Oh! FM THE COMPUTER C MAGAZINE パソコン マガジン		当月 10日

(御注意)

当月発売の号より前に遡ってからの申し込みは出来ません。

バックナンバーは、お近くの書店でお申し込み下さい。

この欄は、加入者あての通信にお使いください。

切り取らないで郵便局にお出しください。

送り先				
お名前	フリガナ	性別 男・女	年齢	ご職業
ご住所	フリガナ 〒			
お電話	ご自宅	勤務先		

申込書			
THE COMPUTER 定期購読	新規申し込み	継続申し込み TC NO.	年間 7,200円
Oh!/PC 定期購読	新規申し込み	継続申し込み PC NO.	年間11,440円 6ヶ月 5,720円
Oh!/X 定期購読	新規申し込み	継続申し込み X NO.	年間 6,720円
Oh!/FM 定期購読	新規申し込み	継続申し込み FM NO.	年間 6,720円
月刊情報処理試験 定期購読	新規申し込み	継続申し込み JS NO.	年間 8,160円 6ヶ月 4,080円
C MAGAZINE 定期購読	新規申し込み	継続申し込み CM NO.	年間11,760円
パソコンマガジン 定期購読	新規申し込み	継続申し込み PM NO.	年間 6,960円
Beep メガドライブ 定期購読	新規申し込み		年間 5,760円
通信欄			

この払込通知票は、機械で使用しますので、下部の欄を汚さないよう特に御注意ください。また、本票を折り曲げたりしないでください。　(郵政省)

[第7話]

スキーは豪華に?

TAKAHARA HIDEKI 高原 秀己

秋も残りわずか。

いよいよスキーシーズンの到来，ってとこである。ぼくもここ数年，1シーズン2，3回はスキーに出かけている。しかし，なかなか上達しない。スコア110のゴルフのほうがまだサマになっていると思うほどなのだから始末におえない。わずか10分で頂上から高速滑走して一気にふもとまで着いてしまうという芸当など，間違ってもできないのである。

しかも悪いことに，やっていて，さして楽しいとは感じられないのもどうにも困ったところである。決してつまらないと思いながらやるのではないにせよ，

「これほど楽しいスポーツはないぞ。スキーこそレジャーの極み」

という友人の感覚には，ほど遠い。

ではなぜ，スキーに行くんだ？　ということになるのだが，これはもう，世の中全体が冬になるとスキー，スキーと大騒ぎしていて，ぼくもしなくてはいけないような妄想に囚われているからにほかならない。

「主体性がない，情けない」なんて声も聞こえてきそうだが，冬の週末に新宿なんかに行ってみると，もう大変。スキーに行く人たちで街はあふれかえっていて，西口方面は駅前から新宿中央公園の南側まで，スキーバスが無限に並んでいる。丸の内なんかも同様だし，果ては大手町や虎ノ門なんかのビジネス街でもスキー板を抱えた，場違いな若い人がウヨウヨしている。

こうなると，ひとりだけ行かないと，なにか社会に参加していないのではないか？とか，いけない生き方をしているのではないか？　と思ってしまったりもする。

まあここまでいうのはオーバーで，スキーのどこが面白いのかを探っているところ，といえばいいのだろうか。

一般的に最近のスキーの特徴といえば，よりぜいたくに，よりファッショナブルに，という点につきるだろう。行き先は北海道，さらにはカナダ，ヨーロッパ。ウエアもパステルカラーあり，蛍光色あり，今年はタウンジャケット感覚の渋めのものありと，年々にぎやかになってきている。

苗場だ，白馬だ，妙高だ，という東京から近いところは，混んでいることもあって，あまり行っても自慢にならない。自慢にならないだけならいいが，まともに滑れないとあっては問題なのだ。

苗場なんかはひどいもので，昨シーズンは280万もの人が集まってしまったとか。1日2万人の割合だから，すさまじい。実際に行ってみると，ゲレンデは人の洪水，状態がちょっと悪い場所になればコブごとに渋滞，スピードを出して滑ってきた人がコブにとりついている人とぶつかって大騒ぎ，といった状態。とにかく滑る場所を探すので，もう精一杯なのだ。

これは苗場だけのことではない。万座，白馬などもシーズン100万人級だし，蔵王なんかだとリフトで数十分並ぶ，なんてことも珍しくない。

その点，北海道はさすがに違う。ゲレンデ自体が広いこともあるが，平日なんかだと，ゲレンデのうち視界に入る部分にはわずか5人，という余裕。上手な人でも下手な人でも，近場よりははるかにいいコンディションで滑れるというわけだ。

「そんなのみんなわかっているんだから，ドバッと人が押し寄せて同じことになるはずだろう」

という人もいるのだが，さにあらず。

なにせ北海道とあって，空路が中心。飛ぶ飛行機の数は決まっているのだから，いわゆる“衝動スキー”組は行けない。

しかも札幌ルートにせよ旭川ルートにせよ，足以上に宿に限度があって，パックツアーなんかも自然に制限がかかってしまう状態になっているという。

北海道にいるということもあって気分だけでも全然違うのだが，環境，混雑状況までいいとあっては，人気は出るはずである。

この北海道スキーをもう一歩，ゴージャスにしたのが流行しはじめている海外スキーだ。なにしろディスカウント合戦の旅行代理店業界とあって，6日間で12万円ちょっと，というカナダスキーツアーまでお目見えしている。

まあこれはシーズン外れゆえの特別価格ではあるのだが，それにしても15万円前後でカナダに行ってスキーができるのだから，恐ろしい話である。

近場で4万円なら6万円出して北海道に，6万円出して北海道なら15万円出してカナダに，というぜいたく感覚，たいしたことがないように思えるのは自然なのだろう。倹約家の人でも近場2回を北海道1回に，ということなら問題はないのだろうし。

と，ここまで書いて改めて感じるのだが，スキーとはいっても，メンバーとか夜の部の活動内容（？）によっても，楽しさたるや全然違うことはいうまでもない。実際，「スキー合宿」と「スキーパーティ」とは天と地ほど違うのだ。

Oh!X INDEX'90

特　集

特別企画

THE SOFTOUCH

GAME REVIEW

AFTER REVIEW

GAME REVIEW (SPECIAL REVIEW)

SPECIAL REVIEW

シリーズ全機種共通システム

連載・シリーズ

知能機械概論　お茶目な計算機たち

猫とコンピュータ

X-OVER　NIGHT

大人のためのX68000

X-BASICプログラミング調理実習

X68000マシン語プログラミング

C調言語講座PRO-68K

ようこそここへC言語

機種別活用・プログラム

イベント／ギャラリー

製品紹介

INFORMATION

NEW PRODUCTS

TFTカラー液晶ディスプレイ搭載
AX386LC
シャープ

AX386LC

シャープは，鮮明表示の10インチTFTカラー液晶ディスプレイを搭載した32ビットラップトップ型AXパソコン「AX386LC」を発売する（12月15日予定）。

表示部には10インチTFT（薄膜トランジスタ）カラー液晶ディスプレイ（640×480ドット，64色中16色表示）を採用し，鮮やかなカラー表示が可能となっている。

本体のほうもクロック周波数20MHzの32ビットCPU80386DX（ノーウェイト），2Mバイトのメモリ，100Mバイトのハードディスク（平均アクセスタイム17ms），16ビットのPC/ATバスを採用したハーフサイズ拡張スロット（AX仕様準拠）を1スロット，およびモデムなどが内蔵できる内部専用スロットを1スロットを搭載し，デスクトップマシンに相当する性能を実現している。

価格は1,490,000円（税別）。

〈問い合わせ先〉
シャープ㈱　☎03(260)1161,06(621)1221

48/24ドットデュアルモードヘッド搭載
XP-2000
セイコーエプソン

セイコーエプソンではデュアルモードヘッドを搭載した48ピンドットマトリクス漢字プリンタ「XP-2000」を発売した。デュアルモードヘッドによって高品位印字実現のための48ドットモードと高速印字のための24ドットモードの切り換えが可能になっている。価格は196,000円（税別）。

また，同社ではターミナルプリンタの普及価格帯におけるラインアップを一新した。発売されたのは，シリアルインパクトプリンタ「VP-960/1600/2600」，熱転写プリンタ「AP-600/900」，インクジェットプリンタ「HG-4000」の計7機種。各製品とも従来機種に比べて低価格を実現しながらも，印字スピードのアップなどの機能強化が図られている。

標準価格（すべて税別）

VP-960	116,000円
VP-1600	123,000円
VP-2600	143,000円
AP-600	62,800円
AP-900	92,800円
HG-4000	196,000円

〈問い合わせ先〉
セイコーエプソン㈱　☎0266(58)1705
エプソン販売㈱　☎03(377)7001

マルチメディア対応TFTカラー液晶ディスプレイを開発
シャープ

シャープは独自の画像処理技術と新しい構想に基づくデジタル駆動システムの確立により，640×RGB×480ドット，1670万色表示可能の高速高精細マルチメディア対応10型および14型TFTカラー液晶ディスプレイを開発した。

NTSCカラー入力やRGBセパレート入力などのアナログ信号をデジタル階調に高速変換する画像処理部と，高速高集積のデジタル多階調ドライバLSIを搭載したTFT-LCDパネル部とで構成されており，オーディオビジュアル用の自然色動画表示とコンピュータコミュニケーション用の鮮明なマルチカラー表示の共用が可能な新しいディスプレイとして使用することができる。

〈問い合わせ先〉
シャープ㈱　☎03(260)1161,06(621)1221

14型TFTカラー液晶ディスプレイ
ハンディステッカーマシン
ステカ
ローランドディー.ジー.

ローランドディー.ジー.では，オリジナルステッカーの製作が誰にでも手軽にできるインテリジェントカッティングマシン「ステカ」を11月20日に発売する。

ステッカーにしたいロゴやイラスト，レタリングをハンディコピーの感覚でサッとなぞり，ステカシートと呼ばれる専用粘着シートを本体に差し込めば，フルオートで原稿のアウトラインがカッティングされる。ステカシートはカラフルなカラーバリエーションの屋外用，屋内用に加えて，アイロ

ステカ

ンプリント用シートもラインアップされ，Tシャツやトレーナーなどにも使用できる。

本体はグレーとホワイトの2色が用意されている。価格は75,000円（税別）。

〈問い合わせ先〉

ローランドディー.ジー.㈱

☎03(5687)1770

電子手帳用カード新3機種
PA-3C20/30/31
シャープ

シャープ電子手帳用ICカード3機種がソフト開発会社より新発売される。

・詰め将棋カード「寅詰」

新進気鋭の棋士，田中寅彦八段が選んだ挑戦問題200問を収録。

㈱スキャップトラスト

標準価格7,500円（税別） 1月発売

・ロードランナー

人気アクションパズルゲーム「ロードランナー」の電子手帳版。全80ステージ。

㈱ナグザット

標準価格7,200円（税別） 発売中

・パズニック

あの「パズニック」が電子手帳で。128面＋アレンジモードの“グラブニック”80面。

㈱タイトー

標準価格5,000円（税別） 12月発売

〈問い合わせ先〉

シャープ㈱　☎03(260)1161，06(621)1221

32ビットIPU
MC68340
モトローラ

MC68340はM68000ファミリマイクロプロセッサとインテリジェント型データハンドリングペリフェラルを，シングルチップ上で集積した32ビットのインテグレーテッド・プロセッサ・ユニット（IPU）である。IPUはHCMOSプロセスを駆使してインプリメントされた完全スタティック動作可能なユニットで，低消費電力動作が可能。スタンバイモードも備えている。CPU32コアはMC68000およびMC68010とユーザーオブジェクトコードでコンパチブルであり，多くのMC68020の追加命令およびアドレッシングモードをも提供する。高速乗算，除算，およびシフト命令も備えているので多くのアプリケーションで性能を向上させることができる。例外処理機能も強化されておりリアルタイム動作で威力を発揮する。さらに，MC68020のパイプライン処理およびプリフェッチの概念も継承されている。

〈問い合わせ先〉

日本モトローラ㈱　0120-068030

INFORMATION

ローランド主催
第3回デスクトップ・ミュージック「力作」コンテスト

1988年に第1回を実施し，今回が3回目となるこの大会は，いわば“デスクトップミュージシャン”の日頃の成果を発表してもらうための場である。コンテストの結果は1991年2月に開催するコンピュータを使った音楽の祭典「デスクトップ・ミュージック・サウンド・パーティ」にて発表される予定。

○募集期間：'90年11月1日～'91年1月10日

○応募方法：有名パソコンショップ，楽器店，大会事務局に備えつけのコンテスト応募用紙を使用。3.5，あるいは5インチフロッピーディスクを添付。

○応募宛先：〒101　東京都千代田区神田須田町2-11　ローランド㈱内　第3回デスクトップ・ミュージック「力作」コンテスト事務局

○応募資格：プロ，アマ，年齢，性別など不問

○応募作品：5分以内。ジャンルは問わない。オリジナル/既成の楽曲どちらも可。ただし，既成の楽曲については作者とタイトルを明記

○賞品：グランプリには50万円相当，準グランプリには30万円相当のローランド製品が進呈される。コンテストの審査機材は下記のとおり

パソコン	NEC　PC-9800シリーズ シャープ　X68000シリーズ 富士通　FM TOWNSシリーズ
音源モジュール	CM-64　（ミュージ郎） CM-32L　（ミュージ郎Jr.） MT-32　（ミュージくん） CM-32P ローランドピアノ　HPシリーズ ※ただし，パソコン本体の内臓音源およびパソコンの拡張スロットに挿入する増設音源は使用しない。
使用ソフトウェア	自作ソフトウェアを含め，一切の制限はありません。
使用インタフェイス	ローランド　MPUシリーズ シャープ　CZ-6BMI システムサコム　SX-68M 富士通　FMT-602＋FM-60-401/FM-401に限ります。

〈問い合わせ先〉

コンテスト事務局　☎03(251)2903

F1情報をネットで

パソコン通信「EYE-NET」の番組（メニュー）に「フジテレビF1情報（有料）」が新設される。いま人気のF1レースに関する情報を提供するこの番組では，予選第1戦，予選第2戦，決勝レースについてそれぞれ1本ずつ計3本のファイルをレースが行われた翌月曜日に掲載する。ピットウォークの裏話やレース予想なども掲載される予定。情報料金は1ファイルあたり250円。

〈問い合わせ先〉

フジミック　☎03(358)0591

コンピュータ図書フェア
書泉グランデ5階

書泉グランデでは「OSから拡がるGUI，ネットワークの世界」と題したコンピュータ図書フェアを12月1日から来年の1月31日まで行う。

○OSの世界－UNIX、MS-DOS，OS/2など

○GUIの世界－X-Window，Windows，OSF/Motif，Open Lookなど

○ネットワークの世界－LAN，Netware，TCP/IPなど

○コンピュータの基本－アルゴリズム＋データ構造，コンパイラ，C言語，PASCAL，Lispなど

以上の書籍を中心にコンピュータ関係の和書，洋書が取り揃えられている。

〈問い合わせ先〉

書泉グランデ　☎03(295)0011

FILES Oh!X

このインデックスは，タイトル，注記——筆者名，誌名，月号，ページで構成されています。何を食べてもおいしい季節になりました。いまのうちにいっぱい食べて，寒い冬に備えて力を蓄えておきましょう。

参考文献
I/O　工学社
ASAHIパソコン　朝日新聞社
ASCII　アスキー
コンプティーク　角川書店
テクノポリス　徳間書店
ポケコンジャーナル　工学社
POPCOM　小学館
マイコン　電波新聞社
マイコンBASIC Magazine　電波新聞社
LOGIN　アスキー

一般

MZシリーズ

X1/turbo/Z

NTT版の贈る“未来はバラ色サイバー色”シリーズの最新刊である，とでもいいたくなるくらい啓蒙色の強い本を出しているNTT出版であり，「サイバーテクノロジー」もそんな香りをたたえている。本書も8人の筆者の原稿＋監修者の月尾氏と筆者との対談という形式で，テクノロジーについて述べていく。内容はアークヒルズで行われている「アーク都市塾」の要約であり，編集は編集工学研究所で装丁は戸田ツトムで刊行はNTT出版となれば目を通さないわけにはいかない。

情報産業をテーマに西和彦氏や松井隼氏（ぴあ総合研究所社長）が語り，アーティフィシャルリアリティをテーマに，武光裕氏がシミュレーションについて，笹田剛史が都市計画とCGAについて，河口洋一郎氏がCGアートについて語る。最後は情報編集をテーマに，電視遊戯大全の著者である石原恒和氏と編集工学研究所の松岡正剛が語る。コンピュータに代表されるテクノロジーを使って新しい試みをしている人たちの考えというのは非常に興味深く，旧態然とした社会の端で興っている文化に触れるには格好の書だ。（K）

セイバーテクノロジー　月尾嘉男ほか8人著　月尾嘉男監修　NTT出版　☎03(435)1212　A5版　214ページ　2,400円

弘幸，マイコンBASIC Magazine，11月号，159-160pp.

X68000

▶NEW SOFT
新着ゲーム，サイバリオン，ジェミニウイング，アクシス～FZ戦記～，ナイアス，Misty Vol.5を紹介。——編集部，LOGIN，19号，14-29pp.
▶最新ゲーム徹底解剖!!
対戦版シムシティーに挑戦。シムシティーは，本来コンピュータと対戦するゲームであるが，遊び方にひと工夫加えて人間対人間のルールを考えた。ほかにウルティマVの攻略法，X68000オリジナルRPGのラグーンの紹介。——編集部，LOGIN，19号，136-143，166-167pp.
▶X68000新聞
熱血高校ドッジボール部サッカー編，アクシス～FZ戦記～，遊撃王II，エメラルドドラゴンなどのほか，新作の情報紹介。——編集部，LOGIN，19号，226-231pp.
▶NEW SOFT
アクシス～FZ戦記～の攻略法のほか，発売予定のニューラル・ギア，ハイドライド3SV，エメラルドドラゴンを紹介。——編集部，LOGIN，20号，16-26pp.
▶最新ゲーム徹底解剖!!
新着RPGラグーンの攻略法第2回。——編集部，LOGIN，20号，154-157pp.
▶X68000新聞
待望のC compiler PRO-68K ver2.0のほか，新着ゲーム“アトミックロボキッド”“パロディウスだ！”“機甲師団”“ハイドライド3SV”などを紹介。——編集部，LOGIN，20号，252-255pp.
▶GAMING WORLD
サイバリオン，熱血高校ドッジボール部サッカー編の紹介。——編集部，テクノポリス，11月号，18-19pp.
▶新作ゲーム先取り Soft Flash
“パロディウスだ！”“アトミックロボキッド”“F15ストライクイーグルII”など，発売予定のゲームを紹介。——編集部，テクノポリス，11月号，29p.
▶攻略おすすめゲーム
アクションRPG“ラグーン”を攻略。——編集部，テクノポリス，11月号，56-58pp.
▶おにおんのアルゴリズムを見切ったぞ!?
カードゲームのアルゴリズムを解説。サンプルとしてX-BASICでのばばぬきゲームを掲載。——編集部，テクノポリス，11月号，118-122pp.
▶ゲームがオレを呼んでいる！
X68000オリジナルのアクションRPG，ラグーンの攻略法。——たかはぴ，POPCOM，11月号，68-71pp.
▶WE ARE THE X68000 WORLD
熱血高校ドッジボール部サッカー編，ナイアス，アクシス～FZ戦記～，サイバリオン，遊撃王II，ハイドライド3SVを紹介。——編集部，POPCOM，11月号，88-91pp.
▶ミュージック・パビリオン
“働く男（ユニコーン）”のミュージックプログラム。——編集部，POPCOM，11月号，167-170pp.
▶X68000SPIRITS
期待の“パロディウスだ！”をはじめ，熱血高校ドッジボール部サッカー編，ハイドライド3SVを紹介。——編集部，コンプティーク，11月号，246-247pp.
▶誌上公開質問状
CZ-603Dにスピーカーをつなぐことができるか？　付属ワープロの「改行幅表示」とは何か？——多田太郎，マイコンBASIC Magazine，11月号，92p.
▶NEW PRODUCTS
X-BASICをコンパイルする際に威力を発揮するデバッキングツール「XBAStoC CHECKER」を紹介。——編集部，マイコンBASIC Magazine，11月号，94-95pp.
▶Mars Knows！
びしばし弾を撃って敵をやっつけていくだけ！　シューティングゲーム。——高橋秀之，マイコンBASIC Magazine，11月号，161-163pp.
▶爆風ポスト2
爆弾を使って手紙をポストに入れるゲーム。——まてりある，マイコンBASIC Magazine，11月号，164-166pp.
▶トリオ　ザ　パンチ
データイーストの同名のゲームミュージックプログラム。——石田勇，マイコンBASIC Magazine，11月号，193-194pp.
▶チャレンジ！　PDS＆同人ソフト
PDSや同人ソフトを紹介する新コーナー。今月はX68000用PDS“OH！ HAJIKI FINAL”“仙人ゲーム”を紹介。——佐久間亮介・やんま，マイコンBASIC Magazine，11月号，234-235pp.
▶AVプログラミング講座
スプライトプログラミング実践編。実際にスプライトを動かすテクニックとスプライトの衝突判定について解説する。——仲田津宏，ASCII，11月号，321-328pp.
▶AV STRASSE
C compiler PRO-68K ver2.0とXBAStoC CHECKER PRO-68Kの2本の内容を紹介している。——編集部，ASCII，11月号，361-364pp.
▶個性派パソコンシリーズ
「キミはとってもオシャレな遊び人」と銘打ってX68000の概要とその性格について紹介する。鳥居勉氏のインタビューも同時に掲載。——荻窪圭，ASAHIパソコン，11.1号，116-121pp.
▶GAME REVIEW
遊撃王II，ラグーン，サイバリオンの3本についての評価記事。——MUNEPI♪・桃子・あゆかわさつみ，マイコン，11月号，193-203pp.
▶なんでもQ＆A
XBAStoC CHECKERとは何か，プリンタのタイムアウトに関する質問，シャープの新型ディスプレイの仕様を掲載。——編集部，マイコン，11月号，398-399pp.
▶GAME BOX
ポピュラスプロミストランド，機甲師団，ジェミニウイングを紹介——市原昌文・吉沢正敏，I/O，11月号，130-132pp.
▶SQUARE-400
ボード上の駒をすべて消し去る知性と気力のパズルゲーム。——佐藤敏孝，I/O，11月号，173-175pp.
▶COPYX & CTRL
拡張版COPY命令と，FDDイジェクト制御のキーを定義するプログラム。——(は)，I/O，11月号，176-184pp.
▶SOFT BOX
バージョンアップされたC compiler PRO-68Kを，変更点を主体に解説。——L & M，I/O，11月号，205-207pp.
▶SCASM
PC-E500/550/1480U/1490UのCPU，SC62015のクロスアセンブラ。データ転送はRS-232Cを介して行う。——小笠原博之，I/O，11月号，161-172pp.

ポケコン

PC-E500
▶MonMon & Pikoron
貧乏なモンモンがピコロンの宝を盗む。アクションゲーム。——小川章，マイコンBASIC Magazine，11月号，168p.
▶ガンバレ　セイビイン2
整備員さんのパズルゲーム。——町野稔，マイコンBASIC Magazine，11月号，169p.

PC-E500/E550/1480U/1490U
▶金融商品損得計算プログラム
利息や複利のデータにそって商品の利率と利回りを算出する。——東浦丞示，ポケコンジャーナル，11号，92-93pp.
▶WOT
相手より先に4つの石を集めろ！　アイテムや武器の豊富な対戦型ゲーム。——IItom，ポケコンジャーナル，11号，67-75pp.
▶北京
“上海”風味の神経衰弱ゲーム。——CUE，ポケコンジャーナル，11号，76-77pp.

PC-1350
▶DRAGON WARRIOR
勇者に奪われた洞窟を奪い返せ！　龍が主人公のRPG。——せとけん，ポケコンジャーナル，11号，88-90pp.

心の社会
皆さまお馴染みミンスキー教授の，邦訳が待たれていた『心の社会』である。「心の社会」というのは「心がたくさん小さなプロセスからできているという考え方」のこと。決して情報処理や人工知能の本ではないし，心理学や社会学の本でもない。心というものに理系なアプローチをしたミンスキー教授の集大成だ。500ページ以上の大作だが，専門用語に頼る部分もなく，読みやすいレイアウトなので非常にありがたい。名著。（K）
マービン・ミンスキー著　安西祐一郎訳　産業図書　☎03(261)7821　A5版　574ページ　3,800円

WORLD ATLAS
マガジンハウスのHot Dog Pressに連載されていたいとうせいこう氏のWORLD ATLASがやっと単行本になった。紙ではなく精神にマッピングされた世界地図とA～Zまでのあやしい辞書形式エッセイ。値段が1,990円というのもあやしい。面白いのは氏のものの見方である。ぬぼーっと生きていては気づかないところに気づかせてくれる，といった感じだ。言葉って面白いと思う。この文体が気にいらない人以外にはお薦め。（K）
いとうせいこう著　太田出版　☎03(359)6262　四六版　260ページ　1,990円

最近，フォーマット済みのフロッピーをよく買ってくるのですがせっかくフォーマットしてあるのにもう一度X68000でフォーマットしなおさないとSYS B:やCOPY COMMAND.X B:してもHuman68kが立ち上がるディスクが作れません。これってなんでなんですか？

長崎県　楠井　富男

「MS-DOSデータフォーマット済み」，「できるFD98初期化済」，「PC-98フロッピーDOSフォーマット」，「MS-DOS FORMATTED FOR PC-98」……最近はMS-DOS用フォーマット済みフロッピーディスクがいろいろなメーカーから出ていますね，私も手元にあるフロッピーだけでこんなに種類がありました。なかにはPC-9801用でありながら対応機種にX68000の名前を載せX68000の写真（なぜかキーボードとディスプレイだけなんだけど）がパッケージに印刷されているものまであって私のようなものぐさな人間には天の恵みといえましょう（そんなごたいそうなもんかい）。

ご存じのようにPC-9801用MS-DOSでフォーマットされた5インチ2HDはHuman68kと同じく1024バイト/セクタ，8セクタ/トラック，154トラックという設定でフォーマットされていますので，そのままHuman68kのデータディスクとして使うことができるわけです。しかし残念なことに，どれも起動用のディスクとして使うためにはSYS.XでHuman68kを転送するだけではだめで，もう一度フォーマットしなおさければならないのですね。せっかくフォーマットしなくてすむと思って買ってきたのに……，なぜこんなことになるんでしょう？

ここではX68000が電源を入れられてCOMMAND.Xが立ち上がるまでを見ていきましょう。

まず，X68000の電源が入れられます。で，いきなりHumanが起動する……わけではありません。まず，電源が入った時点では68000MPUにスタックアドレスと最初に実行するプログラムの入っている番地を入れてまず68000を動かさなくていけないのです（どうだめまいがしてきただろう）。

で，68000が動くのですが，そのときにプログラムの番地を入れましたね。つまりここではまだディスクから読み込んで動き出すのではなくX68000上にプログラムがなくてはいけないのです。X68000の場合はROM上に動き出すためのプログラムが載っています。これをブートストラップローダといいます。

で，ブートストラップローダにはなにが書かれているかというと，諸々の初期化を行ったあとに“ディスクの先頭，トラック0サーフェス0セクタ1からの1セクタ分のプログラムを$2000に読み込みそのプログラムを実行せよ”と書かれています。

いよいよこのプログラムによってディスク上のプログラムが読まれます。ディスク上の最初に動くこの部分はIPLと呼ばれるものでこれがHuman.sysをディスク上から探してメモリ上に引っ張ってきて実行するという役目を持っているのです。そしていよいよHuman.sysが動き，command.xが動く……という構造になっています。

ではいったいフォーマット済みのディスクではなにがいけないのでしょう？

それはIPLなのです。というのはFormatしたときに書き込まれてしまうので他機種でフォーマットしてしまうとX68000用のIPLが書き込まれないのでこのようなことになってしまうのです。つまり立ち上がらなかった原因はHumanのIPLが入っていなかったからだった，というわけなのですね（ちなみに市販のフォーマット済みディスクでは0しか書き込まれていなかったり，PC-9801用のIPLが書き込まれていたり謎のデータが書き込まれていたりする）。

新たにX68000用のIPLを登録するにはディスクの再フォーマットが必要です。といっても，全体を初期化する必要はなく，

format /c

（FATとディレクトリだけの初期化）では時間もかからずIPLもきっちり書き込まれますのでお手軽にIPLを組み込みたいという人はこの/Cスイッチを使うのがいいでしょう。ただ，当然ですがディスク上のプログラムは消えますから使うときには気をつけてくださいね。

X68000のカタログなどを見ると，テキスト画面は1024×1024（表示768×512）ドット，65536色中16色となっているはずなのにいくらやってもテキストでは4色しか出ない。これはサギだ！

岡山県　倉本　仁

……いきなりサギといい切ってしまうのも凄い話ですが……。もちろんX68000はハード的にはカタログに書かれているとおりきっちり16色出すことができます。ただHuman68kではそのように使っていないというだけの話なのです。

ではその隠れてしまった12色はどこにいってしまったのでしょうか。実はこれパレットを細工することによって毎度お馴染みのソフトウェアキーボードとマウスカーソルに使われているのです。

で，プログラマーズマニュアルの図にあ

リスト1

```
 1:                 .include        doscall.mac
 2:                 .include        iocscall.mac
 3:                 .text
 4:                 .even
 5:
 6: gosuper:
 7:                 clr.l           a1
 8:                 IOCS            _B_SUPER
 9:                 move.l          d0,a6
10:
11: IniLDI:
12:                 move.b          #16,d0
13:                 lea             plttbl,a1
14:                 move.l          #$e82200,a0
15:
16: movdat:
17:                 move.w          (a1)+,(a0)+
18:                 subq.b          #1,d0
19:                 bne             movdat
20:
21:                 bsr             DrwBar
22:
23: extdos:
24:                 move.l          a6,a1
25:                 IOCS            _B_SUPER
26:                 DOS             _EXIT
27:
28: DrwBar:
29: *Draw Color Bar
30:                 move.l          #$e08000,a0
31:                 clr.b           d0
32: drwlop:         bsr             drwvrm
33:                 lea             4(a0),a0
34:                 addq.b          #1,d0
35:                 cmp.b           #16,d0
36:                 bne             drwlop
37:                 rts
38:
39: drwvrm:         movem.l         d0-d2/a0,-(sp)
40:                 move.b          d0,d1
41:                 move.b          #4,d2
42: loptxt:         move.b          d1,d0
43:                 and.b           #1,d0
44:                 beq             ndtxt
45:                 bsr             writevrm
46: ndtxt:          lsr.b           #1,d1
47:                 add.l           #$20000,a0
48:                 subq.b          #1,d2
49:                 bne             loptxt
50:                 movem.l         (sp)+,d0-d2/a0
51:                 rts
52:
53:
54: writevrm:
55:                 movem.l         d0/a0,-(sp)
56:                 move.b          #24,d0
57: loopBar:
58:                 move.l          #$ffffffff,(a0)
59:                 lea             $80(a0),a0
60:                 subq.b          #1,d0
61:                 bne             loopBar
62:
63:                 movem.l         (sp)+,a0/d0
64:                 rts
65:
66:                 .even
67:                 .data
68:
69: plttbl:
70:         dc.w    %00000_00000_00000_0    *パレット0の内容
71:         dc.w    %00000_00000_10000_0    *       1
72:         dc.w    %00000_10000_00000_0    *       2
73:         dc.w    %00000_10000_10000_0    *       3
74:         dc.w    %10000_00000_00000_0    *       .
75:         dc.w    %10000_00000_10000_0    *       5
76:         dc.w    %10000_10000_00000_0    *       .
77:         dc.w    %10000_10000_10000_0    *       .
78:         dc.w    %00000_00000_00000_0    *       .
79:         dc.w    %00000_00000_11111_0    *       .
80:         dc.w    %00000_11111_00000_0    *       10
81:         dc.w    %00000_11111_11111_0    *       .
82:         dc.w    %11111_00000_00000_0    *       .
83:         dc.w    %11111_00000_11111_0    *       .
84:         dc.w    %11111_11111_00000_0    *       .
85:         dc.w    %11111_11111_11111_0    *       15
86:
87:
88:                 .end
```

るようにテキスト画面は文字のパターンを描くところが4枚あります。で，その左下に“各ビットごとのテキストパレットにアドレス”と書いてありますね。

この図ではよくわからないかもしれませんね（シャープさんごめんなさい）。ここでいっていることは，簡単に説明すると文字のパターンが，

T3に描かれていたら8を
T2に描かれていたら4を
T1に描かれていたら2を
T0に描かれていたら1を

それぞれ足した数のパレットの色を出す，というふうになっているわけです。

つまり，T3とT0に文字のパターンが描かれていたら，

8＋1＝9

で9番のパレットにある絵の具を使って文字のパターンを出す……と考えてもらえばいいのです。

さて，テキストの文字とマウスカーソルなのですが，Human68kではT0とT1に文字が描かれT2，T3にマウスカーソルやソフトキーボードが描かれるようになっています。さっきの考え方でいくと文字を描いた上にソフトウェアキーボードを描くと文字とキーボードが重なった部分は変な色で表示されてしまうのではないか，と思うでしょう。しかしそこはパレットの魔術。実はHumanではパレットが，

0	黒
1	黄
2	青
3	白
4～7	マウスカーソルの枠の色
8～15	マウスカーソルの地の色

と設定されているのです。これならばたとえば白い文字の上にマウスカーソルの枠があっても，

T0＋T1＋T2＝7
（マウスカーソルの枠の色）

ということになって文字の上にマウスカーソルが乗っているという重ねあわせが簡単にできてしまうのですね。

というわけで，IOCSではサポートされていませんがマシン語で直接I/Oをいじってパレットの I/Oに16色分色を設定してやればちゃんと16色出せるんです。調子にノリついでにサンプルプログラムまでアセンブラで作ってしまいました。参考にしてくださいな。ちなみに，

リストの上段　……パレットの設定
中段　……カラーバーの描き込み
下段　……パレット用の色データ

になっています（色のデータはBRGとテキストVRAMの対応がそのままになるようになっています）。　（古村　聡）

初心者なもんで10月号83ページに載ってる毎日（毎曜日）違う曲を鳴らす方法がわかりません。チェックするっていうのとチェック結果をエラーコードで返すっていうのがわかんないんですが。できればサンプルリストを紹介してほしい……。お願いします。

福岡県　梶谷　太郎

チェックするというのは，今日が何曜日かということを調べることです。チェック結果をエラーコードで返すというのがわかりづらいかもしれません。プログラムはOSから呼び出されてなにかの処理をして戻ります。その際に，正常に終了したかどうかをOSに知らせるのです。この終了コードのうち0～255までの部分は起動したユーザープログラムで利用することが許されています。

終了日曜日を0，月曜日を1……と数字に置き換えて考えると，0から6ですべての曜日を表せますよね。この数値をエラーコードとして扱って，バッチプログラムでプロセス終了コードに応じた処理をさせる，ってことです。Human68kユーザーズマニュアルのCOMMAND.Xおよびバッチ処理コマンドのIFの項を参照してください。

このプロセス終了コードを指定するもっとも簡単な方法はC言語プログラムなどでメイン関数の戻り値として指定してやることでしょう。ここではBASICで書いた場合を例に挙げます（要Cコンパイラ）。

曜日を得る処理（day.bas）は次のようになります。

```
10 int i
20 str week="日月火水木金土"
30 i=instr(1, week, day$)¥2
40 end
```

きわめて簡単ですね。このままではなにも出力しませんが，かまわずBC.XにかけてC言語プログラムにします。エディタで変換したプログラムを読み込み，“b_init( )”という行を削って最後の“b_exit(0)”の部分を“_exit(i)”に書き換えて，

```
cc day.c /W
```

でコンパイルしてください。できあがったday.xを，

```
echo off
day
if exitcode 1 copy sun.opm opm
if exitcode 2 copy mon.opm opm
        ⋮
if exitcode 7 copy sat.opm opm
echo on
```

のようなバッチファイルで利用すればできあがりです。

＊　＊　＊

続いて先月のアフターケアです。先月号でワープロの文書を復活する方法を紹介しましたが，Cコンパイラのバージョン2についてくるデバッガはバージョンが2.00となってプログラムサイズも大きくなったので先月号の説明のとおりにしてもうまく動きません。

それから，先月のプログラムはデバッガのバージョン1.01の使用を前提にしているのですが，システムの使用状況によっては，説明どおりにやってもうまくできない場合があります。そんなときはデバッガ（バージョン不問）でサーチする方法が効果的ですので，うまくいかなかったら，とにかくサーチしてみてください。先頭番地さえわかればこっちのもんです。　（影山　裕昭）

質問にお答えします

日ごろ疑問に思っていること，どんなことでも結構です。どんどんお便りください。難問，奇問，編集室が総力を上げてお答えいたします。ただし，お寄せいただいているものの中には，マニュアルを読めばすぐに回答が得られるようなものも多々あります。最低限，マニュアルは熟読しておきましょう。質問はなるべく具体的に機種名，システム構成，必要なら図も入れてこと細かに書いてください。また，返信用切手同封の質問をよく受けますが，原則として，質問には本誌上でお答えすることになっていますのでご了承ください。なお，質問の内容について，直接問い合わせることもありますので，電話番号も明記してくださいね。

宛先：〒108 東京都港区高輪2-19-13
NS高輪ビル
ソフトバンク株式会社出版部
「Oh! X質問箱」係

FROM READERS TO THE EDITOR

先月は実際は暑いのに涼しいとか書いたけれど，今月は本当に涼しいぞ。どうだ，まいったか。でも，こういう季節は風邪が流行るから気をつけなくては。そのためには温かくして寝るのがいちばん。さあ，さっさと寝よ寝よ。

◆「マシン語カクテル in Z80's bar」は，どうしちゃったんですか？　あのコーナーの登場人物みんなでワイワイやっているのが読んでいてとても面白くて気に入ってたんですけど終わっちゃったんですか？　それならすごく残念です……。
澤田　光彦(18)北海道

いやいや，そう早合点しないでください。10月号はお休みしただけです。どうぞご心配なく。

◆来月，防衛大の入試を受けに行きます。ところで，Oh!X編集部に防衛大出身の人はいますか？（いたらスゴイですね）
川田　剛(18)大阪府

いません。そもそも，みんな身体弱いですから。ゴホゴホ。

◆1990年5月より買っています。X68000を買ったのはもっと前だったのですが，それまでは忙しくて触るひまがなかったのでした。ところで，アンケートハガキにある“好きな石”とは？　なにか意味があるのでしょうか。とりあえず好きな鉱石を書いておきました（黄鉄鉱の結晶，真四角のが好き）。この石を見た時は天然物とは思えないほどきれいな正四面体をしていて，しばらく見入ってしまいました。IC（石）でいえばMC68000系（特に68040？）かな。X68000にもっと速くなってほしいし。「ダイヤ」もいいかな，硬い絶縁物だし。
城戸　吉巳(25)滋賀県

「ダイヤ」はいいですよね。給料3カ月分とかいわれると困るけど。

◆私はパソコンのことはほとんどわかりません。だから，好きな石はなーに？　という質問に真剣に「安山岩と花崗岩かなあー」と答えてしまいました。主人に話さず黙ってアンケートハガキを出していたら，私は間違いなく質問のところにマヌケな答えを書いていたでしょう。それではこれからもがんばってください。
森本　幸子(24)千葉県

だいじょうぶ，「ウケ狙いのつもりで」ICの名前を書いてきた人がたくさんいましたから。

◆「あなたの好きな石は何ですか？」，……よく考えるといまはない。1979年頃，8080Aで自作を始めたので，「思い出の石」というのはたくさんあるが，いまはハードを一から作ることはありえないので石の好き嫌いをいうレベルではない。CPUは68系が好きだったのでX68000を選んだのだ。
「私の思い出の石」
一瞬にして燃えた1個6,000円くらいだったVDGの6847。大容量メモリを作るつもりで大量購入したら2114が安く出回りだして使わずにオクラ入りした1個1,200円くらいのSRAM2112。書き込み器の失敗でダメになったEPROM2708。
平山　謙司(40)福岡県

石に歴史あり，ですね。

◆10月号は音楽特集ということでLIVE in '90がいままでにまして気合いの入ったいい曲があってよかった。自分はこのコーナー命の人間なのでどんどんいい曲が載るのを期待しています。ちなみに，いままでの曲の中でいちばんよかったのは「RYDEEN」です。この曲は感動しました。
城田　雅弘(17)群馬県

「RYDEEN」は最初の馬の足音がいいですよね。えっ，それはなかった？

◆先日，バンド仲間のライブを見に行って，とび入りで歌わされた。ひさしぶりのスポットのせいか右目がとても痛くなった。そして，曲の後半，フレーズのちょうど盛り上がりのあたりで涙がポトリ……。突然のことに客席もそのバンド仲間もびっくりしたのだと思う。なんともいえないキンチョーが私のほうに伝わってきた。とっさに，「ちょっと思うところがあって。ごめんなさい……」などとしっかりとコメントしてステージを降りた。あの雰囲気じゃ，まさか本当のことなど言えないよネ。それにしても皆さん，女の涙にはご用心ですヨ!!
矢吹　準子(24)福島県

よくとっさにそんなコメントが出てきましたね。「ちょっと玉ねぎが目に……」とかいったらさぞかし場がシラけただろうなあ。

◆会社の人に「X68000がほしいの」といったら，「PC-9801のほうが絶対にいいよ〜」といわれてしまった。そこで，「だって，PC-9801ってばカッコワルイんですもん」といったら，返す言葉もなく引き下がった。X68000がPC-9801に勝ったぞ。
田中　泰代(19)神奈川県

返す言葉がなかったんじゃなくて，あきれてしまっただけかも。

◆本棚にOh!Xが5冊並んでいます（そのとなりにミュシャの画集が……）。先日，なんとか前期試験も終わり，大型台風のお陰で遅れたOh!Xを読んでいると，父が「嫁入り道具のパソコンは何がいいか」といいました。しばらく何も言えない私。まだぢゅうはちだよおう。でも，“X68000 HYPER−HD”(笑) あたりかな……なんて思うこともあります。次の新作は一体何だろう？
岩瀬　貴代美(18)福岡県

嫁入り道具にパソコンも買ってもらえるとはなんて幸せ。でも，結婚するころには次のX68000どころか，次の次のX68000とかが出てたりして。

◆今日は（9月17日）朝からいいことしたんだぜぇ。一日一善ってやつだな。駅前で傘がなくて困っている高2の女の子と一緒に学校まで行ってあげたんだ。「どお？」って聞いたら，「すいませぇ〜ん」ってな具合。しかし，名前聞いておくんだった。くそ。私はその子と同じ

◀富田　祐樹　東京都
最近，8周年とか100号達成とかもあったんで，「おめでとう」ハガキはいっぱい来ていたんですけど。ほんと，ありがたいことです。

◀木下　義崇　佐賀県
あいかわらず大人気のソーサリアンですね。女の人は見たところ魔法使いのようですけど。髪の毛の感じがなかなかいいですよ。

学校に通う3年生である。ふん，一日一善だよ。それだけ！　　　　　　高橋　政秀(17)東京都

イマイチ押しが甘いなあ。でも，人に親切にするのはいいことです。

◆感熱紙はスパイ用アイテムである。

証明）必要条件　証拠となる文章が残らない

十分条件　熱，日光であっという間に読めなくなる

証明終わり

びえー！　うっかり日向に置いておいた感熱紙が真っ茶っ茶になっちゃたよー！　LHarcの使い方とか，ROGUEのキー操作とかイロイロプリントしてあったのに。リボン高いし，そのうえいまとなっては取り寄せ(CZ-8PC2用なので)だし……。もっと根性のある感熱紙はないのかー！　ドットインパクトプリンタがほしーよー。

松本　康裕(23)広島県

根性がありすぎて字がプリントできないというのも困りますが。

◆僕はサイボーグ，またはアンドロイドを作りたいと思っています。友人は電子工学科で頭脳を作ってくれるそうです。友人の失敗に備えて，誰か脳を提供してくれませんか？　ボディーができた暁には入れてあげましょう。めざせ！ダニール・オリバー！　うーん，マッドなハガキだ……。　　　　　　枝松　樹(20)愛媛県

サイボーグに脳の提供ねえ。死んだあとだったら考えないこともないなあ。いまやると痛いもん。

◆日本国内に生息する蝶は250種くらいいるのですが，この夏，私の標本箱にあらたに1種が加わり，145種が揃いました。ほとんどすべて自分の足で歩き捕虫網をふるって（少々はずかしいんですが）採集し展翅をして標本にしたものです。CARD-PROを使い，現在，日本産蝶類のデータベースを作っています。MacintoshのHyperCardなら電脳図鑑が作れるのになあ……。

斎藤　光一(40)埼玉県

蝶も最近あまり見かけなくなりましたね。さみしいかぎりです。昔は取って食うほどいたんですけどねえ。

◆数あるパソコン雑誌の中でも内容がいちばんまともでいい。西川善司氏が面白い。余談でよく編集部内の狂っている部分とか書かれているが，そういうメンバーでどうやってちゃんと編集しているのか不思議だ。でも，面白ければなんでもいい。　　　　堤　雅秀(22)神奈川県

信じてくれー。オレは決して狂ってなんかいなーい。うおー。ガンガンガン。

◆音楽の世界というものは入ってみると非常に面白いものであるとは思うのです。しかし，楽譜が読めないとか，楽器のひとつもできないとか，作曲なんて人間のできることではないとか，そういったことが頭を過ぎるたびにその世界に入ることをためらってしまうことが，ままあるのではないでしょうか。コンピュータミュージックはそういった壁を打破するための，もっとも手っ取り早い方法たりうるとは思うのですが。

船山　竜士(21)埼玉県

◀田沼　康弘　東京都
うーん，想像とは恐ろしい。あのフラッピーがこんなふうになるとは。これじゃあ敵のほうを応援してしまいますね。カワイイもん。

◀伊藤　圭一　東京都
2人とも目がうつろですが，かえってそれで目を引かれてしまいました。寂しくて鋭いなにかを感じてしまいました。

大人になってから，「ああ，子供の頃からなにか楽器を習っていればよかった」とか思うんですよね。楽器やってるとモテるだろうし。

◆シムシティー，9つ目のシナリオ。それは……，まず空き地がなくなるまで開発する（災害なしモードで）。そして，金をある程度貯めたら予算カット。災害連発で町を完全に破壊する。そうすると，一面焼け野原の土地になるので，そこから町を作る。シナリオ名は戦後の復興。町を破壊する前の金の貯め具合でレベル設定をします。一度くらいはやる価値があると思いますが，どうでしょう？　福永　浩司(19)大阪府

それで，町が成長してきたらまた破壊するということを繰り返すんですか。人間の歴史そのものだなあ。

◆某大手電機メーカーのコンピュータH/W技術者から実家の酒屋にデューダして早半年。最近は論理回路の飛び回る夢も見なくなってしまい，とてもさみしいものです。転職前はそれでも日曜日などに設計でもして……と思っていましたが，実際はそのようなひまもなく，睡眠の補給と買い物の荷物持ちで終わってしまいます。せめて貴誌の記事を見て欲求不満を……などとも考える今日この頃です。　林　将智(28)大阪府

日曜日はやっぱり睡眠の補給だけで終わりますよね。まあ，休めないときよりはいいけれど。

◆Oh! X LIVE in '90に初めて掲載されてから，早2年近くになります。その間いろんなことがありましたが，憧れであった“常連”などとも呼ばれるまでになりました。私ももうすぐ大学4年生就職活動もあって忙しくなります。いまでも十分レポートで大変です。なぜこんなことを書くかというと，そう，もう投稿はよそうと考えて……は，まったくいません。ミュージックプログラムはすでに私のライフワークと化しているのです。次は“スペシャリスト”の域に達することができるように気合い入れます。

安藤　正洋(21)青森県

がんばってください。

◆どうもすみません。アンケートハガキの「あなたの愛機は」のHD(　MB)のところを，僕はいままでフロッピーのメディアと勘違いし，2HD(1MB)と書いておりました。ごめんなさい。

柴田　和久(18)東京都

あらあら。

◆Oh!Xを買ってひと通り目を通してから，次号予告を見て「おお，来月はこういうのがあるのか」と思って楽しみにしています。しかし，いざ買う時には「おお，今月はこういうのをやっているのか」と，すっかり忘れてしまっているわけなんです。　　　　阪本　泰博(20)大阪府

本屋さんに行く前にちゃんと予告を読んで予習していきましょう。

◆熱血高校ドッジボール部野球編というのを作ってほしいなあ。クロスプレイでの乱闘！　必殺ピッチャー返し！　バットを砕く魔球！　絶対すごいと思う。　　　　木全　克徳(21)京都府

なんか，熱血高校というよりは「あばっち野球軍」になってしまいそうですね。

◆この間，レンタル電話の申し込みをしたら断られてしまった。料金はクレジットカードから落ちるため，まずクレジットカードの審査があるのだが，残念ながら私の預金口座は残高がマイナス50万円ぐらいであるため，作れなかったのである。いつになったら通信専用電話が持てるのだろうか。ちなみにマイナスになったのはX68000を買ってしまったせいなのだ。一児の父より。　　　　　矢崎　慎一(35)東京都

かわいそうな日本のお父さん。ううっ。

◆電脳倶楽部Tシャツである。コード表が逆さまになっているのがいかがわしいが，よぉ〜く考えてみるとこれはとっても便利なものではないか！(特許取ったんですか？)

伊藤　洋司(19)茨城県

あのコード表にはそのほかにも，その人の体の大きさにあわせて文字の大きさが変わるという機能もついているんですよ。

◆ビデオボードの購入を見合わせていたけれど，10月号の改造記事を見て買い，作ってみました。別にノイズもなく素人工作にしてはソツなく出来上がり喜んでいます。

伊規須　一男(40)福岡県

日曜大工ならぬ，日曜工作。よかったですね。

◆そうだったのか！「OPMというファイルでFM音源演奏可能」というのは，ああやってOPMファイルを作るということだったのか。それを知らずにいままで生きてきたのか。ああ，10月号買ってよかった。　安部　一馬(23)福岡県

そのまま知らずにいたら一生後悔するところでしたね？

◆うちの学校は生徒の99パーセントが進学希望者である(あくまでも希望)。僕はというと残りの１パーセントの人だったりする。だから，夏休みはバイクや車で（免許取ったのは）遊び回り，バイトで金も稼いだ。おかげさまで，すでにダブる確率50パーセント以上である。同学年の先生で僕の名前を知らない人はいないくらい有名である。……こんな僕はどうしたらいいの？　だれか教えてちょうだいな（自分が悪いクセに……)。おっと，バイトの時間だ（自覚のない私)。　菅野　弘治(18)東京都

まだ確率が40パーセントくらい残っているじゃないですか。あとは本人次第ですね。

◆テストの前になると，私のノートを借りにくる人がいる。「これで自分と点が同じか，あるいはよかったら割に合わないな」などと，ふと考え込んでしまう。うーむ。

小野寺　光(20)宮城県

貸した相手が単位を取ってるのに，自分は落としたというのがいちばん悲惨なパターンですね。

◆いやあ，盛況でしたね，バックナンバーフェア。午前中はあまり行動しないのですが，この日（９月20日）ばかりは早起きして行ってきました。着いたのは10時30分ごろ。会場の５階はもう若い人でいっぱい（といっても20人ぐらいですが，開店直後のため店内は閑散としていたのでこのコーナーの人だかりは普通でなかった)。私も探していた1987年度後半の７冊を無事購入しました。しかし，これは少ないほうでひとりで30冊ぐらいひとかかえにしてレジに並ぶ人が多数いたのには驚いてしまいました（重いだろうな)。　岡野　英司(42)東京都

そうだったのか。買いにいけばよかった。くそー。

◆この間，「C compiler PRO-68K ver.2.0」を買いました。噂には聞いていましたが，実際に見てやっぱりすごいなと思いました。あのマニュアル。ちびちび読んでいますが，わからない言葉などがあってなかなかはかどりません。なにやらOh!XではＣ言語の連載が始まったようですが，これぞまさにナイスタイミング。初心者のボクにもよくわかるようにお願いします。

石浦　芳仁(20)東京都

はははは。そうではない。あの連載を始めるためにシャープに圧力をかけてver.2.0を発売させたのだ。ウソ。

◆あの，私まだ子供だからよく知らないんですけど，やっぱり大学生になったら機械語という言葉を習うんですか？　そ，それで，あ，あの機械語っていうのは，どこの国で話されているのですか。機械和辞典というのはあるのですか？　加藤　伸(15)千葉県

そんなもんありません。

◆DōGAが大変。CA68は（あ，私のチームです）人手不足，時間不足，資金不足，睡眠不足で大変です。しかも，仕事(グラフィックデザイナー)も忙しい。デザイナーには時間がないのよ。がちょーん。……。でも，がんばってCGAコンテストには出します。　安藤　優子(22)福岡県

寝ずにがんばって出してください。

◆難しくてわけのわからないページの下のいろいろ書いてあるのを読んでいました。すると自分の書いたのが載っているではありませんか。編集部の皆様，ありがたき幸せにございます。なんか恥ずかしいものですね。ほかの皆様のと比べると見劣りがします。でも，自慢しようと思うのは人間の性ですね。友人にいいふらそうと思いましたが，Oh!Xの存在すら知らない友達ばかりで……。がんばってください。

笹山　克巳(16)石川県

自慢するついでにOh!Xを宣伝するというのがいいでしょう。

◆SLANG＋REDAで開発した作品を手に，某ソフトハウスの入社試験に臨んだところ，なんと内定をいただきました。THE SENTINELのコーナーを支えている方々に感謝しています。いやあ，Oh!Xの読者でよかった。この未熟者の私が合格できたなんてまだ信じられません。

西田　一郎(17)長崎県

いやいや，運がよかっただけですよ。あっ，違う違う。あなたの実力ですよ。

◆10月号の174頁の溝渕さんへ。JRの比較的古い特急用車両のデッキ側の壁（ドアの横）にはちゃんとAC100Vのコンセントが付いています。ただし，プラグが特殊なので変換コードを自作する必要があります。洗面所には電気カミソリ用に普通のプラグの差し込めるコンセントがあるのですが，それを使うとほかのお客さんに迷惑ですので。　津幡　岳弘(23)愛知県

これで，寒い日にコタツを持って電車に乗っても安心ですね。

◆THE USER'S WORKSはいい企画だと思います。ネットをやっていなくてもこのようなソフトにお目にかかれるのは，普通のユーザーにとっていい刺激になっていいと思います。実際，私もそのひとりです。　樋口　雅人(18)福島県

あのコーナーは結構評判がいいんですよね。ああいうものって，なにか手作りの味があっていいですからね。

◆私の通っている学校がある町所沢は，ご存じ西武ライオンズの本拠地である。この町のダイエーでは西武優勝に便乗してバーゲンを催すことになっている(優勝したらの話だけれど)。そして，賑やかな商店街のあるほうではなく，駅を挟んで反対側のほうに２軒のパソコンショップが同じビルの中に存在している。ここは秋葉原でもないのに，だ。そこで激しい価格競争が行われるのだが，とうとう2HDのフロッピーディスクのブランド品が一番安いもので680円にもなった。これは秋葉原にも匹敵するものがある。このまま競争が続けば我々消費者はウハウハものである(けど，ここで安定するだろうな)。はたして２店の運命やいかに？

小川　純一(17)埼玉県

あんまりやりすぎて両方ともつぶれたら，みんなさぞかし困るだろうなあ。

◆知り合いにピアノを10年くらいやっているやつがいる。そいつもパソコンユーザーなので，MMLデータの書き方は知っていたのだが，私が「そのデータだけ見てピアノを弾いてみろ」というと，妙な具合に顔をそむけて言った。「慣れたらできる。……と思う」。

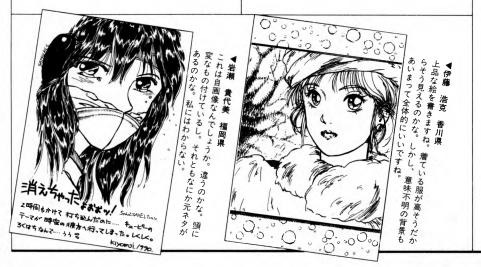
矢部　尚之(17)大阪府

できたらスゴイ。でも，不可能ではないですよね。

◆ついにプログラマとして働くことになりました(まだアルバイトの身ですが)。しかも，ゲームソフトのプログラマです。いままでは市販のゲームにあれやこれやとナンクセをつけていた私ですが，いざ自分で作るとなると，ハードの制約，メモリの制約など目に見えなかったところに問題があることを知り，ゲームに対する見方も「これはどうやっているのだろう？」というふうに変わってきました。パソコンでのプログラミングはあくまで趣味として続けていきたいと思っております。　久保　誠(27)京都府

そのへんがむずかしい問題なんですよね。

◀岩瀬　貴代美　福岡県
これは自画像なんでしょうか。違うのかな。頭に変なもの付けているし。それともなにか元ネタがあるのかな。私にはわからない。

◀伊藤　浩克　香川県
上品な絵を書きますね。着ている服が高そうだからそう見えるのかな。しかし，意味不明の背景もあいまって全体的にいいですね。

趣味を仕事にすると面白くないという人もいれば，好きなことをやってお金をもらえるなんて最高と思う人もいるし。

◆現在，月に十数冊の雑誌を購入している。主に電気，コンピュータ関係の本ばかりである。その他にSFの本も月に数冊買っている。これを1年半続けた結果，ベッドの半分が本箱になってしまった。あと1年たてば，私は本箱の上で寝ることになるかもしれない。

渡部　裕亨(23)福岡県

本を並べてその上に寝ればベッドがいらなくなるじゃないですか。変な夢にうなされそうですけど。

◆僕は6畳間に住んでいるんですが，10月号の福島県の安藤君。4畳半というのは気の毒だが，まだ狭い部屋に住んでいる人も多い。キーボードスタンドや十数台のオーディオ機材，X68000とX1turbo，音源モジュールと雑誌(Oh!X，キーボードマガジン，OPTION，CAR BOY，バリバリマシン等）はまだ我慢する。しかし，どうにもならないっていうのがA1大の製図板だ（マウスマットの代わりにするとGOOD)。友人で下宿にドラフタ（?!）があるやつもいるが，全国の機械科諸君！　メゲずにがんばろうぜ！

佐藤　仁(21)山梨県

ドラフタですか。ドラム缶のフタじゃ大変でしょう。ちがうって！

◆10月号の特集の電子音楽術入門はいくらなんでもよすぎた。ZMUSIC.FNCなんて最近「あったらいいなあ」と思っていたものがポーンと出ていたのだからビックリ。もう，その辺のウン千円もするコンピュータミュージックの解説書のうえをいってます。……とこれだけヨイショしておけば，プレゼントは僕のもの……。あっ，ウソですよ。うん，本当にグッド。

加藤　武史(16)福井県

あの特集は西川善司君が大活躍でしたね。善司先生に励ましのお便りを出そう!?

◆三国志IIで新君主を女性にしてプレイしていると，たまにほかの君主から「わが娘をぜひ貴殿の嫁にしていただきたい」という密書が来たりするが，君主は女なんだけどなあと思って考えてしまいます。しかし，密書の内容が「わが息子をぜひ貴殿の婿にしていただきたい」というものだったら，どうだろうか。

藤原　彰人(20)岡山県

でも，プレイしているのは男である君。うーん，なんかややこしい。

◆今年の夏は古くなった扇風機がよく燃えたらしい。うちのはスイッチを入れても最初は動かない。そこでエルボースマッシュを1発食らわすとしばらくして動き出す。だが，うちのやつは燃えたりはしない。なぜなら35年前の扇風機はすべてアルミと鉄でできているから。いったい世の中進歩しているんでしょうかね？

寺尾　文治(38)岡山県

ということは3才のときから使っているわけですね。なんとものもちがいい。でも，押し入れにしまうときに重そう。

▲見浦　崇　長野県
耳が尖っている。こいつは人間じゃないな。ローブを着ているほうも怪しい。やい，正体を現わせ。でも，Oh!Xを持っているから許してあげる。

◆初めて買った。近々，X68000を買う予定なのでOh!Xを買うことにした。本屋でパラパラと見たときは，活字ばかりで非常にかたくるしそうだったが，よく読んでみると非常に面白かった。これからも買っていくつもりだ。

清野　一男(19)秋田県

毎月着々と読者が増える，Oh!X。

ぼくらの掲示板

●掲載ご希望の方は，官製ハガキに項目(売る・買う・氏名・年齢・連絡方法……)を明記してお申し込みください。
●ソフトの売買，交換については，いっさい掲載できません。
●取り引きについては当編集部では責任を負いかねます。
●応募者多数の場合，掲載できない場合もあります。
●紹介を希望されるサークルは必ず会誌の見本を送ってください。

仲間

★X1ディスクユーザーの皆さん！　X1を見放すのはまだ早い。このたび，CURECではX1ユーザー総集結号と題して，全国のX1ユーザーを対象にディスクマガジンの制作，発行を予定しています。とにかくX1ユーザーのパワーでなにかすごいことをやろうと考えていますので，ひとりでも多くの方の参加をお願いします。とりあえず下記住所へ62円切手同封のうえご連絡ください。折り返し案内状をお送りします。〒488　愛知県尾張旭市東栄町根の鼻5186-40　水野義則

★X68000ユーザーを対象とするサークル「SAKIKO」では会員を募集しています。このサークルではX68000に関する情報提供を中心に活動しています。会員から送られてきた情報を整理してディスクで配布します。誰でも参加できる自由なサークルを目指しているのでよろしく。詳しくは62円切手同封のうえ，下記まで。〒671-12　兵庫県姫路市勝原区山戸241-10　山根邦博（16)

売ります

★MZ-2500用のカラースキャナ・ユーティリティ「SS-SC25C」と，ハンディカラースキャナ「WD-05HS」を4万5千円から5万円でお売りします。〒277　千葉県柏市柏715-11　ジェネパレス柏505-101号　加藤康成（19)

★オムロンのモデム「MD1200AIII」(1200bps，箱，付属品あり）を9千円で。送料は負担してください。連絡は往復ハガキで。〒755　山口県宇部市上宇部中尾　松尾明法（18)

★シャープのプリンタ「CZ-8PG2」を8万円で売ります。6カ月使用，箱，マニュアル，付属品すべてあります。連絡は往復ハガキで。〒562　大阪府箕面市粟生間谷西3-7-9-408　波戸博司(27)

買います

★X68000用の1Mバイト以上の増設RAM（CZ-601Cに接続できるもの，完動，付属品つき）を送料込み1万3千円くらいで買います。連絡はハガキで。〒708-15　岡山県久米郡柵原町八神248　礒山直樹（18)

★拡張I/Oポート「CZ-8EP」を5千円，コンパクトフロッピーディスク「CZ-3FBD」等（3インチ）を1枚200円，ミニフロッピーディスクドライブ「CZ-801FS」を1万5千円で。連絡はハガキで。〒福岡県北九州市八幡西区医生ケ丘1番5-407号　国藤恭正（38)

★MIDIボード「CZ-6BM1」を送料込み1万5千円くらいで。完動品でマニュアル，付属品つきのものを。連絡はハガキで。〒203　東京都東久留米市氷川台2-15-7　佐藤晶（17)

★X68000用数値演算プロセッサボード「CZ-6BP1」を3万円以下で（完動，マニュアル，付属品つき）。連絡は往復ハガキで。〒158　東京都世田谷区上用賀4-7-3　伊東雅子（29)

★MZ-700/1500用純正ジョイスティック「MZ-1X03」を千円で買います。連絡はハガキで。〒399-07　長野県塩尻市片丘10391　古筬一浩（21)

バックナンバー

★Oh!Xの1989年3，4，5月号を送料込み各2千円で。切りぬき以外は可。連絡はハガキで。〒285　千葉県佐倉市城271　伊藤徹（20)

編集室から from E・D・I・T・O・R

DRIVE ON

●「大人のためのX68000」について。さまざまなデータを扱うときに，住所録というある程度固定されたデータから入るのは正しいと思う。また，エディタなどで住所録を作りだすときにまだ見ぬ明日のことを考えて，項目ごとの区切りを統一させるのは賢明であるが，実際においてはそのときのフォーマット（形式）が非常に難しい。その形式次第であとのそのデータの使われ方はがらりと変わる。その最初の取り決めにはセンスが必要であるが（これはほかのジャンルのソフトにもいえる），そこを詳しくやっているのは評価される点である。

長谷川　敦士(17)　MZ-2500, MSX2　山形県

●今回は辞書の強化に興味がありました。他機種の辞書を利用するなんて思いもしなかったのですが，思ったより手軽で今度実行してみたいと思います。X68000にもそろそろ新しいFEPが発売されてもいいころだと思うのですが，10月号を見て「やはりASKも捨てたものじゃない」と改めて思いました。

中川　比呂志(19)　X68000, XICs　東京都

●FM音源ですか。世にMIDIが広がるなか，なにか新鮮なものを感じますね。私が初めて組んだプログラムはミュージックプログラムでした。手軽にできるわりに，うまくいったときの感動はとても大きなものですよね。ただ，音色の設定には熟練を要するのではないでしょうか。そこで私がお勧めしたいのは，いまさらながら「SOUND PRO-68K」です。本当に「いまさら」という気はしますが，FM音源を使いこなしたい人にとってはなかなか頼もしいものですし，使えるものです。これで慣らしていけば，しだいに音色設定の達人になれるのではないでしょうか。15,800円は決して高い買い物ではないように思います。

安井　百合江(16)　X68000PRO　愛知県

●「THE USER'S WORKS」について。同人ソフトは安価であり，市販ソフトとはひと味違った「手作りの味」があるので，紹介記事を載せるのは賛成である。制作する側も反響が高ければやりがいがあるはずだ。

泉　昭彦(20)　X1turbo, PC-E500　東京都

●「ようこそここへC言語」について。基礎面を中心にということですから，ある程度連載が続いてもレベルを上げないでほしいですね。しかし，それでは記事も行き詰まりを見せてしまうでしょう。だから，まったくの初心者だった人がある程度のプログラムなら作れるというレベルまで上がったと思われるころになったら，大きな(もしくは中くらいの)プログラムを数カ月に分けて完成させるというのはどうでしょうか。X-BASIC講座のコラムスのときや，(で) のぱーてぃハンズみたいにです。そういえば昔「ロードランナーで学ぶC言語」とかいう本もありましたね。

髙橋　毅(19)　X68000PRO, MSX2　埼玉県

ごめんなさいのコーナー

11月号　ごめんなさいのコーナー

ZMUSIC.FNCのところで訂正番地が間違っていました。029Aではなく，0299を2Bにしてください。

7月号　ハードウェア工作入門

P.107　図2の回路図と8月号に掲載した図1の実体配線図が異なっているようです。これは製作中に変更点があったためで，回路図は変更前のものです。図1が正しい回路図です。ご迷惑をかけました。

6月号　INTEGRAL X1

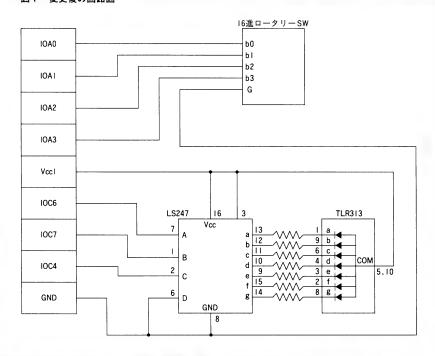
すでにあるファイルと同じファイル名のものをコピーした場合，ファイルサイズが更新されません。下のように訂正してください。

```
0000                 1  ;
0000                 2  ;KAME-DOS BUG-FIX
0000                 3  ;
0000                 4  ;  S-OS REDA
0000                 5
E042 P               6  #ERR7      EQU $E042
D084 P               7  #FNAM1     EQU $D084
E068 P               8  #FNAM      EQU $E068
0000                 9
0000                10  ;
D565                11   ORG $D565
D565                12  ;
D565 C3 30 DE       13   JP BEGIN
D568                14  OPNSK4
D568                15
D568                16  ;
DE30                17   ORG $DE30
DE30                18  ;
DE30                19  BEGIN
DE30 C2 42 E0       20   JP NZ,#ERR7
DE33 21 C8 D0       21   LD HL,#FNAM1+46+22
DE36 11 7E E0       22   LD DE,#FNAM+22
DE39 01 0D 00       23   LD BC,13
DE3C ED B0          24   LDIR
DE3E C3 68 D5       25   JP OPNSK4
DE41                26
```

図1　変更後の回路図

身近になったC言語 1年ぶりのC特集

▼ようやくC compiler ver. 2.0も発売されました。これを機にC言語を導入された方も多いことでしょう。そしてOh!Xでも1年ぶりのC言語特集です (思えば，昨年の特集もXCの新バージョンにタイミングをあわせて企画していたような気が……)。いまやプログラミング言語の中核となったC言語。ちょっとしたファイル操作から本格的な開発まで，あらゆる分野で活用してください。

▼Oh!Xでは誌面作りを手伝っていただく協力スタッフを募集しています。仕事内容は原稿執筆，プログラム開発，投稿チェックなど。希望者は6000字程度の自由論文を添えて「スタッフ募集」係まで連絡してください。

▼さて，ひそかに予告されていたとおり，来月号はディスク (5″2HD) つきでお届けする予定です。なにが入っているかは開けてみてのお楽しみ。

愛読者プレゼントの応募方法

とじ込みのアンケートはがきの該当項目をすべてご記入の上，希望するプレゼント番号をはがき右下のスペースにひとつ記入してお申し込みください。締め切りは1990年12月18日の到着分までとします。当選者の発表は1991年2月号で行います。

10月号プレゼント当選者

1ルーンワース (山形県) 簗瀬信悦 (東京都) 倉持聡 (徳島県) 谷口成広　2ワールドコート (北海道) 谷口有香 (神奈川県) 角井真吾 (大阪府) 堀川英知　3闇の血族 (山形県) 宮下丈司 (東京都) 井上綾子 (岩手県) 泉哲也　4電脳倶楽部Tシャツ (北海道) 佐藤政幸他19名　豆しぼり (北海道) 白戸知巳他9名　えんぴつ (栃木県) 広田義弘他9名　シャープペン (千葉県) 伊藤徹他9名

(敬称略)

以上の方々が当選されました。おめでとうございます。商品に順次発送いたしますが，入荷状況などにより遅れる場合もありますのでご了承ください。

投稿応募要領

- 原稿には，住所・氏名・年齢・職業・連絡先電話番号・機種・使用言語・必要な周辺機器・マイコン歴を明記してください。
- プログラムを投稿される方は，詳しい内容の説明，利用法，できればフローチャート，変数表，メモリマップ (マシン語の場合) に，参考文献を明記し，プログラムをセーブしたテープ (ディスケット) を添えてお送りください。また，掲載にあたっては，編集上の都合により加筆修正させていただくことがありますのでご了承ください。
- ハードの製作などを投稿される方は，詳しい内容の説明のほかに回路図，部品表，できれば実体配線図も添えてください。編集室で検討のうえ，製作したハードが必要な場合はご連絡いたします。
- 投稿者のモラルとして，他誌との二重投稿，他機種用プログラムを単に移植したものは固くお断りいたします。

あて先

〒108 東京都港区高輪2-19-13 NS高輪ビル
ソフトバンク出版部
Oh!X「㋢㋮㋯」係

SHIFT・BREAK

▶前期の成績をもらったあと，友達とボーリングを7ゲームをやる。編集室に来てマーブルマッドネスをやる。「今日は玉ばっか転がしてるね」と，A.T.氏にいわれる。痛む腰をさすりながら，AMIGAの前に座って必死にトラックボールを転がしている僕。でもみんな，前期の成績まで坂道を転がり落ちるように下がっていたのは内緒だぞ。(純)

▶都内では全部の車を駐車場に入れたとしても10万台以上の車が路上にあふれる計算だそうである。こんな状態で駐車違反を2点にしたところで警察の小遣い稼ぎにしかならない。交通事故が増えると真っ先に責められる警察の立場もちょっとだけ同情するが，交差点内駐車などの迷惑度の高いものから取り締まってほしいものだ。(この前1点減ったH.U.)

▶アサルトコンパネの情報どうも有難うございました。ところで，編集室は「闇の血族」が密かなブーム。もう日常会話中でも「んーもうJESUS」とか「-SIGH」とか「はふ」とか「私はね今，BLUEな気分なんだ」とか手を叩いて「Slap！」とか叫ぶ始末。え？　そんなことしているのは私だけ？　うっそぉ。(ワタシデナイワタシ善ジデナイ善ジ)

▶コンビニエンスストアでごちゃごちゃと買い物をしたら，金額がちょうど2,000円 (税金も合わせて) であった。妙に嬉しい。と思ったら，バイクの走行距離が30,000kmを超える瞬間を見逃してしまった。妙に悔しい。ところで，AMIGA購入計画は資金面で挫折して，増設メモリに転んでしまった。いやあ快適快適，転んでもただでは起きないよ。(A.T.)

▶この号の発売の2日後にはもう，初スキー。気がつけば今年も終わりに近い。そしてこの僕も，X68000を買ってしまった。やはり時は無情に流れていくのだ。そんな状況を反映してか，最近何かと忙しい。でも，こんなときだからこそ自分らしさを失わないようにしたいと思う。最近アウトドアが好きだ。(今年はシヌマデスキーのC)

▶ゲームソフトの年末商戦はすごい。特に今年はこれでもかというくらい多くのソフトが発売される。それだけソフトハウスにとっては厳しい戦いとなるわけだ。でも，こういうときって不思議とひとつのゲームが異常に売れちゃったりもするんだよね。あのザナドゥがそうだった。果たして今年は？　ショップへ行って何が売れるか占ってみよう。(S)

▶なるほど，能書きを垂れないで何もできない連中は多い。能書きを先に垂れては必ず転ぶ。ああ，腐った能書きが多くていけないや。能書きがいけないんじゃない。ああ，精神をナメた肉体と，肉体をナメた精神と，世の中にはどちらかしかないのか？　心と体の思考のバランス。心の腐ったやつを殴り倒して独裁したい男のロマン，な秋。(K)

▶バージョン2のLK.XはCASH.Xと相性が悪い。XCのバージョン1やGCCからLK.Xを呼び出すと失敗する。そこで，CC.X (バージョン1) とGCC.XをDIS.Xでソースジェネレートし，電脳倶楽部から手に入れたHLK.Xを呼び出すように改造してやった。これで世界に平和が戻った。それにしてもバージョン2のコンパイル速度は遅すぎると思いません？(KO)

▶この間までは「部屋にクーラーがない」とかいって騒いでいたのですが，涼しくなってきてふと周りを見わたすと暖房器具がなにもない。しいて挙げるなら，布団と2台のコンピュータぐらいでしょうか。4月に上京してきたところだとはいえ，なんとも情けないかぎり。コタツでも買うしかないかな。ちょうど，机もないことだし。(A)

▶最近体力がひどく落ちてきたので「このままでは死んじゃうかも……」と思い，ジョギング用にスニーカーを買ってきた。が，それも「徹夜明けの身体で走ったら心臓マヒで死んじゃうかも……」という心配に代わっただけ。結局そのスニーカーを履いて，家でダンスエクササイズとバーレッスンをやっている今日この頃の私……情けない……。(E.O.)

▶「贅沢を持つ喜び」も捨てがたいが，とりあえずA500を買った。これで3台目，あともう1台は誰かに……，あ，MIDIも買わなきゃ。さて，来月号は特別定価780円だ。そうそう，生ディスクの用意も忘れないように。そんでもって，SX-WINDOWは本当にやってくるぞ。RAMの準備はいいか？　ハードディスクに余裕はあるか？(U)

▶かつては70万円もしたMacintosh SEの相当品が19万8千円。当時みんなの憧れだったMacIIに相当するLC (68020，2MB，256色カラー) もEXPERTより安い。まあ冷静に見積もればそんなものかなという気もするが，X68000のハードは重装備だからそれほど安くはならないだろうな。うーむ。(T)

microOdyssey

私の机にはSS-NETの電話機が載っている。シャープ製だが，子機はない。そのかわり，こいつには立派なRS-232C端子があって，同じ机のX68000とつなげばUNIXの端末になる。つまりSS-NETというのは社内の電話回線を使ってLANを構築できるシステムである。凄いやつだ。目のつけどころがシャープである。

さて，編集部の電話機は交換機が変わったり，引っ越したりで，何回も変わっている。そのたびに，電話機の操作に戸惑い，外線を誤って切り，内線番号表を書き換え，さらには100件近い短縮番号を登録してきたわけだ。

そこで今回は電話の悪口を書く。へんに思想めいたことより，はっきり悪口と決めたほうが書きやすい。まあ電話料金が高いとかいったことは今回は見逃してあげようと思う。

実は，そのSS-NETの電話機にもいささかの不満がある。だって，今どき液晶表示がないんだもの。以前使っていた電話機では，かけた番号を液晶パネルにエコーバックして確認することができた。簡単なことだが，あるとないでは大違い。リダイヤルや短縮番号を使う際にも，相手の番号が表示されるのは便利である。表示がないと，短縮番号を登録しても，実際に相手に電話をかけてみないことにはちゃんと登録されているかさえわからない。これって間抜けな話でしょ。

ところで，東京都では局番が4桁になる。市内通話に8桁の番号が必要なのだ。覚えるのも大変だが，番号を入力ミスする確率も確実にアップする。緑の公衆電話にはカードの残り度数を大きな文字でデジタル表示する。それができるんだったら，エコーバックしてくれよ。と言いたい。後ろの人に相手の番号を見られて困るなら，表示をOFFにするくらい簡単でしょ。うう，きっと考えてないんだな。

じゃあ，表示のある電話機ならいいかというとそうでもない。表示があっても電話番号はエディットできない。市外番号だと10桁にも及ぶのに，最後の1桁を間違えただけで最初からやり直しだ。

ここでちょっと考えてほしい。東京都の電話番号が足りないのはわかる。でもいまある3桁の局番の頭に3をつけて4桁にするというのはなんかおかしい。新しい局番だけ4桁にすればいいのでは？　ところがこれがダメ。電話にはエンドコードというものがないからだ。たとえば，1234-5678という局番4桁の番号作ったとしても，これを局番3桁の123-4567という番号と区別するこできないのである。

もう桁数についてはしょうがない。あとは今後設置される電話機に便利になってもらうしかない。私としては，ある程度の大きさの液晶表示とメモリをつんで電話番号を管理してほしい。登録した番号は別の電話機に転送できるようにしてほしい。メモリカードを公衆電話に使えるようにしてほしい。そうそう，プッシュボタンの配列は電卓と同じにしてほしい。とにかくなんとかしてほしいのだ。

その昔，プッシュホンに変えると計算機能があると聞いて感動したが，それは電話機に電卓機能があるのではなく，電話をかけて計算してもらうサービスのことだった。今はそういう時代じゃない。　(T)

バックナンバー常備店

東京	神保町	三省堂神田本店5F 03(233)3312
	〃	書泉ブックマートBI 03(294)0011
	〃	書泉グランデ5F 03(295)0011
	秋葉原	T-ZONE 7Fブックゾーン 03(257)2660
	八重洲	八重洲ブックセンター3F 03(281)1811
	新宿	紀伊国屋書店本店 03(354)0131
	高田馬場	未来堂書店 03(200)9185
	渋谷	大盛堂書店 03(463)0511
	池袋	リブロ池袋店 03(981)0111
	〃	西武百貨店9F コンピュータ・フォーラム 03(981)0111
神奈川	横浜	有隣堂横浜駅西口店 045(311)6265
	〃	有隣堂ルミネ店 045(453)0811
	藤沢	有隣堂藤沢店 0466(26)1411
神奈川	厚木	有隣堂厚木店 0462(23)4111
	平塚	文教堂四の宮店 0463(54)2880
千葉	柏	新星堂カルチェ5 0471(64)8551
	船橋	リブロ船橋店 0474(25)0111
	〃	芳林堂書店津田沼店 0474(78)3737
	千葉	多田屋千葉セントラルプラザ店 0472(24)1333
埼玉	川越	黒田書店 0492(25)3138
	川口	岩渕書店 0482(52)2190
茨城	水戸	川又書店駅前店 0292(31)0102
大阪	北区	旭屋書店本店 06(313)1191
	都島区	駸々堂京橋店 06(353)2413
京都	中京区	オーム社書店 075(221)0280
愛知	名古屋	三省堂名古屋店 052(562)0077
	〃	パソコンΣ上前津店 052(251)8334
	刈谷	三洋堂書店刈谷店 0566(24)1134
長野	飯田	平安堂飯田店 0265(24)4545
北海道	室蘭	室蘭工業大学生協 0143(44)6060

定期購読のお知らせ

Oh!Xの定期購読をご希望の方は綴じ込みの振替用紙の「申込書」欄にある『新規』『継続』のいずれかに○をつけ，必要事項を明記のうえ，郵便局で購読料をお振り込みください。その際渡される半券は領収書になっていますので，大切に保管してください。なお，すでに定期購読をご利用の方には期限終了の少し前にご通知いたします。継続希望の方は，上記と同じ要領でお申し込みください。

海外送付ご希望の方へ

本誌の海外発送代理店，日本IPS(株)にお申し込みください。なお，購読料金は郵送方法，地域によって異なりますので，下記宛必ずお問い合わせください。

日本IPS株式会社
〒101 東京都千代田区飯田橋3-11-6
☎03(238)0700

12月号

■1990年12月1日発行　定価560円(本体544円)
■発行人　孫　正義
■編集人　橋本五郎
■発売元　ソフトバンク株式会社
■出版事業部　〒108 東京都港区高輪2-19-13　NS高輪ビル
Oh!X編集部 ☎03(5488)1309
出版営業部 ☎03(5488)1360 FAX 03(5488)1364
広告センター ☎03(297)0181
■印　刷　凸版印刷株式会社

満開の電子ちゃん

作：いわいいっぺい
え：岡村祭

電気は？

だから電気は？

「電脳倶楽部の購読を始めたら成績はアップ、部活ではレギュラー、おまけに彼女までできました」という文章を何年も使い回すほど年季が入れば推薦文も本物でしょう。しかし、今までにここからデビューした傑作の数々には目を見はるものがあります。CGに革命を起こした「PIC」や、ファイル管理ツール「F」等を手放すのは、酒をやめるより苦しいに違いありません。これらは当然、品切れなしのバックナンバーでいつでも入手できます。デバッグやバージョンアップのアフターケア（再掲載）も嬉しいですね。

平木敬太郎（福井県）

秋葉原価格でローンができます
電気の街秋葉原で
25年の信用!!

AVCフタバ

☎03(253)7661

〒101 東京都千代田区外神田3-2-3 ☎03-253-7611(代)

今すぐ もよりの電話から	仙　台 022-264-3704	名古屋 052-452-3271	広　島 082-295-6873
札　幌 011-611-5104	新　潟 0252-75-4175	大　阪 06-311-3931	福　岡 092-481-2494

X68000の情報のすべて!(当店はX68000の認定代理店です。お気軽にご相談下さい)

48ドット熱転写プリンター。精密な文字、ハードコピーも可能。

CZ-8PC4……¥ 99,800

AVC特価¥???

お勧めディスプレイコーナー　組合せは自由、価格はお気軽にご相談下さい。

CZ-604D 標準価格¥94,800 AVC特価 ●0.31mmドットピッチ ●2モードオートスキャン ●ステレオスピーカ搭載 ●チルト台同梱	CZ-613D 標準価格¥135,000 AVC特価 ●ドットピッチ0.31mm ●TVチューナー搭載 ●ステレオスピーカー搭載 ●チルト台同梱	CZ-603D 標準価格¥84,800 AVC特価 ●0.31mmドットピッチ ●TVチューナ無し ●3モードオートスキャン ●チルト台同梱
CU-21HD 標準価格¥148,000 AVC特価 ●0.52mmドットピッチ ●21型ディスプレイ ●3モードオートスキャン ●ステレオスピーカ搭載	CZ-605D 標準価格¥115,000 AVC特価 ●ドットピッチ0.39mm ●TVチューナー搭載 ●ステレオスピーカー搭載 ●チルト台同梱	CZ-602D 標準価格¥99,800 AVC特価 ●ドットピッチ0.39mm ●TVチューナー搭載 ●チルト台同梱

●但し消費税(3%)は別途請求させていただきます。●分割回数は3回〜48回まで自由に選べます。

X68000 PERSONAL WORKSTATION EXPERT HD

CZ-612C-BK ……………¥466,000
CZ-602D-BK ……………¥ 99,800

セットでお買上の方に、SX-WINDOW、ジョイカード、"グラデーウス"ディスケット10枚プレゼント!

AVC特価　¥368,000

SUPER HD

80MBハードディスク、SCSIインターフェース搭載!
CZ-623C-TN ……………¥498,000
CZ-613D-TN ……………¥135,000

AVC特価　お電話で……………………

●セットの組合せは自由!広告に出ていない他の機種はお問合せ下さい。

●頭金なし(手軽な電話クレジット)●製品先取り(お支払いは約1〜2ケ月後から)●低金利クレジット(1回の支払いは2,700円以上で3〜48回。ボーナス併用も可)●カレッジクレジット(保証人なし。但し満20歳以上の学生の方)●18歳未満の方(ご両親が代理購入者としてお申し込み下さい)●納期(通常の場合、当社に申込書が到着後1週間以内。特に人気のある商品で品薄の場合、少々納期が遅れることがありますので御了承下さい)●完全保証(すべてメーカー保証書付。アフターケア万全)●全国代引(お届けした者に、代金をお支払いいただく方法です。但し手数料1,000円)

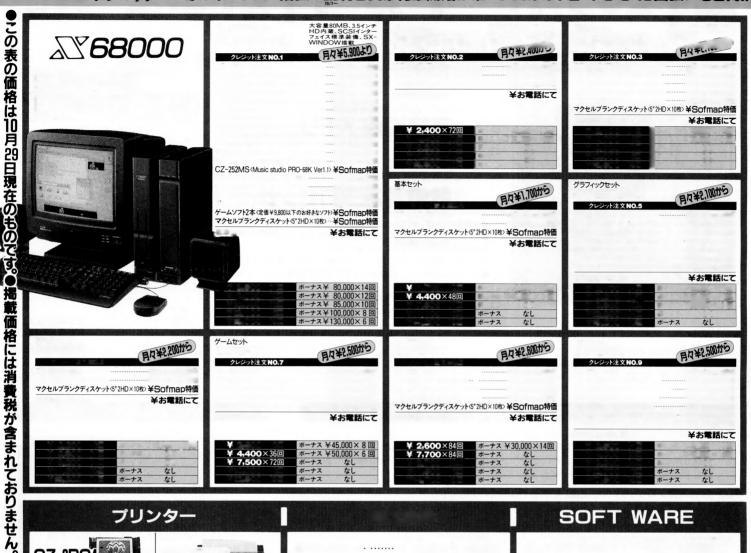
AM10時からPM7時まで受付 日曜・祝日も営業

ACCEEL TOMATO series ICM LAND OMRON メルコ maxell I·O DATA SONY FUJI FILM Caravelle WINTECH

Sofmap 商品は今すぐ！お支払いは9ヶ月後から。

For Computer Communication Age

株式会社ソフマップ　冬のボーナス一括払い金利0（ゼロ）受け付け開始!! ボーナスのみ1·2·4·6·8·10回払いもOK!!

●この表の価格は10月29日現在のものです。●掲載価格には消費税が含まれておりません。

CZ-8PG2
定価¥160,000
¥お電話にて

¥お電話にて

No.1システム

No.1 下取りシステム

お持ちの機種を下取りに出して、新品に買替えようと思っている方、ソフマップに御相談下さい。
買取り価格がどこよりも高く、新品の販売価格がどこよりも安いから、当然どこよりもお得な条件でお買求めいただけます。
又、差額を商品券でお支払いもできます。

No.1 配送システム

1. 到着日指定、夜間配送システム
   お客様のご都合に合わせて配送させていただきます。機種によっては、夜間配送できないものがあります。
2. 代金引換システム（要手数料）
   係員が品物をお届けに行きますので、その時にお支払い下さい。

No.1 クレジットシステム

1. 9ヶ月先からのお支払いOK
   スキップクレジットを御利用になれば支払い開始月を1ヶ月から、最長9ヶ月先までおくらせる事が出来ます。
2. 月々¥1,000からのお支払いOK
   月々のお支払い金額の設定が¥1,000からOK。
3. 84回払いもOK
   お客様のプランに合わせて、1回から最長84回まで支払い回数をお選びいただけます。
4. ステップアップクレジット
   お客様のプランに合わせて、毎月のお支払い金額を徐々に増やしていくシステムです。例えば、1年目は¥3,000、2年目は¥6,000というように、御自由に設定することができます。
5. ボーナス10回払いもOK
   毎月の支払いは0（ゼロ）、ボーナス時のみのお支払いでクレジットが御利用になれます。回数は1、2回の他、4·6·8·10回払いまでOK。
6. カードクレジット
   各種クレジットカードが店頭だけでなく、通信販売でも御利用になれます。詳しくはお気軽にお問い合わせ下さい。
7. カレッジクレジット
   保証人なしで、学生の方でもクレジットが御利用できます。（20歳以上）

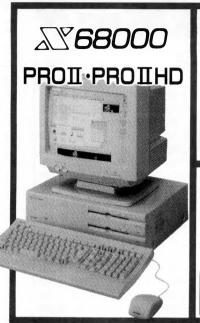
No.1 サポートシステム

1. 初期不良交換期間3ヶ月
   ●万一、お届けした商品が不良の場合、お買い上げ日より3ヶ月以内なら、同等品と即、交換致します。
2. 新品パソコン3年保証
   ●メーカー保証が1年の場合、メーカー保証1年＋マップ保証2年の計3年間の保証になります。
3. 中古パソコン1年保証
   ●中古パソコン本体は、1年間保証致します。（ディスプレイプリンタ等は6ヶ月保証となります）
4. 新品パソコン買取り保証
   ●1ヶ月以内であれば必ず買取り保証金額で、下取り、買取り致します。
5. 永久買取り保証
   ●古くなったパソコン、スクラップ寸前のパソコンでもOK!! どんなパソコンでも、どこよりも高く買い取ります。

ACCEL TOMATO series ICM LAND OMRON 株式会社メルコ maxell I-O DATA SONY FUJI FILM Caravelle.12 WINTECH

下取り・買取り大観迎!!

TV 日本テレビ、TBS、フジテレビ、テレビ朝日、テレビ東京系列でCM放映中!! 直営10店舗

下取り差額のお支払いは、クレジットを御利用下さい。

基本セット PROⅡ-HD 月々¥2,300から

クレジット注文NO.10

- CZ-663C〈本体〉……¥Sofmap特価
- CZ-605D〈15″ドットピッチ0.39〉……¥Sofmap特価
- マクセルブランクディスケット〈5″2HD×10枚〉¥Sofmap特価

標準価格¥511,000 ¥お電話にて

月々	ボーナス
¥2,300×72回	ボーナス ¥30,000×12回
¥4,900×60回	ボーナス ¥20,000×10回
¥8,300×60回	ボーナス なし
¥12,600×36回	ボーナス なし
¥18,100×24回	ボーナス なし

ビジネスセット PROⅡ-HD 月々¥2,300から

クレジット注文NO.11

- CZ-663C〈本体〉……¥Sofmap特価
- CZ-605D〈15″ドットピッチ0.39〉……¥Sofmap特価
- CZ-8PG2〈24ピン漢字ドットプリンター130桁〉……¥Sofmap特価
- CZ-212BS〈BUSINESS PRO-68K〉……¥Sofmap特価
- マクセルブランクディスケット〈5″2HD×10枚〉¥Sofmap特価

標準価格¥739,000 ¥お電話にて

月々	ボーナス
¥2,300×72回	ボーナス ¥50,000×12回
¥6,100×48回	ボーナス ¥50,000×8回
¥9,700×84回	ボーナス なし
¥12,000×60回	ボーナス なし
¥14,400×48回	ボーナス なし

データベースセット PROⅡ-HD 月々¥2,300から

クレジット注文NO.12

- CZ-663C〈本体〉……¥Sofmap特価
- CZ-613D〈15″ドットピッチ0.31〉……¥Sofmap特価
- CZ-8PG1〈24ピン漢字ドットプリンター80桁〉……¥Sofmap特価
- CZ-226BS〈CARD PRO-68K〉……¥Sofmap特価
- CZ-220BS〈DATA PRO-68K〉……¥Sofmap特価
- マクセルブランクディスケット〈5″2HD×10枚〉¥Sofmap特価

標準価格¥748,800 ¥お電話にて

月々	ボーナス
¥2,300×84回	[illegible] ¥45,0[illegible]×[illegible]回
¥5,500×60回	[illegible] ¥40,000×10回
¥9,800×84回	ボーナス なし
¥13,500×54回	ボーナス なし
¥18,600×36回	ボーナス なし

基本セット PROⅡ 月々¥3,200から

クレジット注文NO.13

- CZ-653C〈本体〉……¥Sofmap特価
- CZ-605D〈15″ドットピッチ0.39〉……¥Sofmap特価
- マクセルブランクディスケット〈5″2HD×10枚〉¥Sofmap特価

標準価格¥401,000 ¥お電話にて

月々	ボーナス
¥3,200×60回	ボーナス ¥20,000×10回
¥5,300×84回	ボーナス なし
¥7,800×48回	ボーナス なし
¥10,000×36回	ボーナス なし
¥14,400×24回	ボーナス なし

通信セット PROⅡ 月々¥1,800から

クレジット注文NO.14

- CZ-653C〈本体〉……¥Sofmap特価
- CZ-613D〈15″ドットピッチ0.31〉……¥Sofmap特価
- CZ-8PG1〈24ピン漢字ドットプリンター80桁〉……¥Sofmap特価
- MD-24FS5〈通信モデム2400BPS〉……¥Sofmap特価
- CZ-257CS〈Communication PRO-68K Ver2.0〉……¥Sofmap特価
- マクセルブランクディスケット〈5″2HD×10枚〉¥Sofmap特価

標準価格¥620,600 ¥お電話にて

月々	ボーナス
¥1,800×60回	ボーナス ¥50,000×10回
¥5,400×36回	ボーナス ¥60,000×6回
¥9,000×72回	ボーナス なし
¥12,100×48回	ボーナス なし
¥15,400×36回	ボーナス なし

プリントセット PROⅡ 月々¥2,200から

クレジット注文NO.15

- CZ-653C〈本体〉……¥Sofmap特価
- CZ-605D〈15″ドットピッチ0.39〉……¥Sofmap特価
- CZ-8PC4〈48ドット熱転写プリンタ〉……¥Sofmap特価
- CZ-221HS〈NEW Printshop PRO-68K〉……¥Sofmap特価
- CZ-235GS〈グラフィックライブラリVOL.1〉……¥Sofmap特価
- CZ-236GS〈グラフィックライブラリVOL.2〉……¥Sofmap特価
- マクセルブランクディスケット〈5″2HD×10枚〉¥Sofmap特価

標準価格¥538,200 ¥お電話にて

月々	ボーナス
¥2,200×60回	ボーナス ¥40,000×10回
¥4,700×48回	ボーナス ¥35,000×8回
¥7,100×84回	ボーナス なし
¥9,800×54回	ボーナス なし
¥13,500×36回	ボーナス なし

●掲載の商品以外にも多数取り扱いしておりますので、お気軽にお問い合わせ下さい。又、商品在庫は毎日変動しますので、品切れの際は御予約承ります。

どこよりもお得 今しかない! 差額クレジットOK!!

下取り差額は随時変動します。

高額下取り差額表

この他の商品についてもお電話でお気軽にお問い合わせ下さい。

お送りになる方、又は直接東京店に来られる方 0120-110-894
直接大阪店に来られる方 06-641-8801
商品の先り先 〒101 千代田区外神田1-8-3 野水ビル3F ソフマップ買取りセンター

あなたが今、欲しい機種(新品) / あなたが今、お持ちの機種

下取り機種	SUPER-HD CZ-623C CZ-613D	EXPERT Ⅱ CZ-603C CZ-605D	EXPERTⅡHD CZ-613C CZ-613D	PROⅡ CZ-653C CZ-605D	PROⅡ-HD CZ-603C CZ-605D
	交換差額	交換差額	交換差額	交換差額	交換差額
CZ-652C CZ-602D	¥308,000	¥135,000	¥238,000	¥95,000	¥135,000
CZ-602C CZ-602D	¥268,000	¥95,000	¥198,000	¥55,000	¥95,000
CZ-611C CZ-611D	¥270,000	¥97,000	¥200,000	¥57,000	¥97,000
CZ-601C CZ-601D	¥318,000	¥145,000	¥248,000	¥105,000	¥145,000
CZ-600C CZ-601D	¥323,000	¥150,000	¥253,000	¥110,000	¥150,000
CZ-880C CZ-880D	¥440,000	¥267,000	¥370,000	¥227,000	¥267,000
PC-9801VX21 PC-KD854N	¥323,000	¥150,000	¥253,000	¥110,000	¥150,000
FM-TOWNS-2 FMT-DP531	¥398,000	¥225,000	¥328,000	¥185,000	¥225,000

高額買取価格表

商品名	高額買取
X68000 モニターセット	
X68(CZ-662C+CZ-600D/601D)	¥250,000
X68(CZ-662C+CZ-611D/612D)	¥260,000
X68(CZ-652C+CZ-600D/601D)	¥210,000
X68(CZ-652C+CZ-611D/612D)	¥220,000
X68(CZ-623C+CZ-602D)	¥360,000
X68(CZ-623C+CZ-605D)	¥380,000
X68(CZ-623C+CZ-613D)	¥390,000
X68(CZ-623C+CZ-603D)	¥345,000
X68(CZ-623C+CZ-604D)	¥350,000
X68(CZ-612C+CZ-600D/601D)	¥290,000
X68(CZ-612C+CZ-611D/612D)	¥300,000
X68(CZ-611C+CZ-600D/601D)	¥235,000
X68(CZ-611C+CZ-611D/612D)	¥245,000
X68(CZ-603C+CZ-602D)	¥255,000
X68(CZ-603C+CZ-605D)	¥270,000
X68(CZ-603C+CZ-613D)	¥280,000
X68(CZ-603C+CZ-603D)	¥215,000
X68(CZ-603C+CZ-604D)	¥225,000
X68(CZ-602C+CZ-600D/601D)	¥240,000
X68(CZ-602C+CZ-611D/612D)	¥250,000
X68(CZ-601C+CZ-600D/601D)	¥195,000
X68(CZ-600C+CZ-600D/601D)	¥190,000

業界No.1の低金利

支払回数	1	3	6	10	12	15	18	20	24	30	36	42	48	54	60	66	72	78	84
他社金利	3	4	5	7	9	10	12	13	16	19	21	25	28	31	35	—	—	—	—
Sofmap金利	2.0	3.0	4.0	5.5	5.5	8.5	11.0	11.0	11.5	16.0	16.0	20	21	26	27	33	35	39	42

お支払い方法

1. 代金引換システム
係員か品物をお届けに行きますので、その時にお支払い下さい

2. クレジット
お電話て支払い回数、支払い開始日、ホーナスの有無をおっしゃって下さい
こちらからクレジット用紙をお送り致しますので、ご記入・ご捺印の上ご返送下さい
商品到着後、御指定の口座から自動引落しとなります

3. 銀行振込
お電話て御注文の上、下記振込先へ電信扱いてお振り込み下さい
ご確認後、たたちに商品をお送りします 振込手数料はお客様負担となります

振込先
東京秋葉原店
三和銀行秋葉原支店(普)1012131
口座名義 ㈱ソフマップ

電話受付時間/年中無休 ●平日 AM11:00〜PM8:00 ●日・祭日 AM10:00〜PM7:00

店頭に直接来られる方は 東京03-258-3156 大阪06-647-0562

通信販売をご利用の方は 0120-110-833 大阪06-633-7224
FAX.03-253-4290
札幌011-865-7030 横浜045-311-3441 広島082-222-0604
仙台022-268-3405 金沢0762-21-7045 福岡092-752-0044
新潟0252-22-6139 名古屋052-332-2117 高松0878-34-8833

24時間テレフォンサービス 03-258-7910
フリーダイヤル 商品発送のお問合わせ 0120-08-0113
フリーダイヤル 故障・修理のお問合わせ 0120-11-0292

〒101 東京都千代田区外神田3丁目15番6号小暮末広ビル1F
〒556 大阪市浪速区日本橋5丁目7番17号ソフマップビル

SHARP

パソコン本体から周辺機器まで品数取り揃え

大特価セール実施中!!

型名	品名	正価	特価
UE-1D02	14インチカラーディスプレイAXシリーズ	158,000	特価
UE-1D03	15インチカラーディスプレイAXシリーズ	123,000	特価
UE-1E03	5"FDインターフェイスカード AXシリーズ	28,000	特価
UE-1E02	AX286LICカードI AXシリーズ	45,000	特価
UE-1E04	S-RNインターフェイスカード AXシリーズ	70,000	特価
UE-1P01	136桁漢字プリンタ AXシリーズ	268,000	特価
UE-1P02	速136桁漢字プリンタ AXシリーズ	550,000	特価
UE-1P05	6桁漢字水平プリンタ AXシリーズ		特価
UE-1P04	136桁漢字プリンタ AXシリーズ		特価
UE-1P03	80桁漢字プリンタ AXシリーズ		特価
UE-1R04	2M RAMボード AXシリーズ	180,000	特価
UE-1R03	2M RAMボード AXシリーズ	100,000	特価
UE-1R05	張グラフィックボード AXシリーズ	92,000	55,000
UE-1R01	2M RAMボード AXシリーズ	300,000	特価
UE-1R06	ROM ボード AXシリーズ	32,000	25,600
UE-1R02	2M RAMボード AXシリーズ	100,000	特価
UE-1U01	X 286L スロットBOX AXシリーズ	5,000	4,000
AX286D-FH4	MZ-8306A	458,000	特価
AX286D-F	MZ-8302A	278,000	特価
AX286L-F	MZ-8352A	428,000	特価
AX286L-FH3	MZ-8353A	598,000	特価
AX386-FH4	MZ-8706A	1,100,000	特価
AX386S-FH4	MZ-8706B	780,000	特価
AX386-F	MZ-8702A	860,000	特価
AX386S-F	MZ-8702B	590,000	特価
AX386-FH8	MZ-8707A	1,280,000	特価
AX386S-FH8	MZ-8707B	960,000	特価
CE-126P	ポケコンプリンター	17,800	13,800
CE-124	ポケコンカセットインター	4,500	3,600
CE-120P	ポケコンプリンター	24,800	21,800
CE-123P	ポケコンプリンター	19,800	17,800
CE-140F	ポケコンフロッピーディスク	49,800	40,300
CE-140T	ポケコンRS-232Cコンバーター	9,800	8,800
CE-159	ポケコン RAM 8K	35,000	4,200
CE-158	ポケコンレベルコンバター	39,800	31,300
CE-1600E	ポケコンディスクインターフェイス	19,800	17,800
CE-1601M	ポケコン RAM 64K	45,000	30,000
CE-161	ポケコン RAM 16K	50,000	3,800
CE-1650F	ポケコン DISK	9,800	8,800
CE-1600P	ポケコン プリンター	69,800	59,800
CE-1600F	ポケコン フロッピードライブ	39,800	34,800
CE-1600M	ポケコン RAM 32K	32,000	16,000
CE-201M	ポケコン RAM 8K	18,000	3,000
CE-202M	ポケコン RAM 16K	35,000	6,000
CE-203M	ポケコン RAM 32K	32,000	7,000
CE-T800	ポケコン RS-232Cコンバター	12,800	11,800
CZ-300F	X13"マイクロフロッピー	79,800	9,000
CZ-31F1	300F増設フロッピー	59,800	7,000
CZ-501H	X1増設用ハードディスクユニット	258,000	特価
CZ-503F	CZ-830増設ドライブ	49,800	30,000
CZ-520F	2HD/2DDミニフロッピードライブ	118,000	70,000
CZ-6BG1	GPIBボード	59,800	47,800
CZ-6BP1	数値演算ボード	79,800	63,800
CZ-6BC1	FAXボード	79,800	65,000
CZ-6BU1	ユニバーサルI/Oボード	39,800	33,800
CZ-6BM1	MIDIボード	29,800	23,800
CZ-6BEIA 1M	X68000増設RAMボード	38,000	19,500
CZ-6BEIB 1M	X68000増設RAMボード		19,500
CZ-6BEI 1M	X68000増設RAMボード	35,000	29,500
CZ-6BN1	スキャナーボード	29,800	25,300
CZ-6BE4	X68000増設RAMボード	138,000	特価
CZ-6BF1	RS-232C 増設ボード	49,800	42,300
CZ-6BE2 2M	X68000増設RAMボード	79,800	特価
CZ-6EB1	拡張 I/O BOX	88,000	69,800
CZ-6ST1	チルトスタンド	5,800	3,500
CZ-6SD1	システムラック	44,800	38,000
CZ-6TU G.B	RGBシステムチューナー	33,100	26,500
CZ-822C	X1G MODEL30	118,000	35,000
CZ-820C	X1G MODEL10	69,800	16,800
CZ-888C	XI TURBO Z3	169,800	95,000
CZ-8BGR2	グラフィックボード X1	14,800	3,000
CZ-8BF1	FDインターフェイス	14,800	11,500
CZ-8BK2	漢字ROM	19,800	16,800
CZ-8BM2	232C マウスセット	19,800	16,800
CZ-8BE2	320K外部メモリー	29,800	25,300
CZ-8BR1	立体映像セット	39,000	33,800
CZ-8BV2	カラーイメージボード	39,800	32,000
CZ-8BO1	FDインターフェイス	14,800	8,000
CZ-8BM2	モデムユニット	49,800	39,800
CZ-8EB3	振張 i/obox	33,800	28,000
CZ-8LM1	232cケーブル	7,200	6,000
CZ-8LM2	232cクロスケーブル	7,200	6,000
CZ-8NJ1	ジョイカード	1,700	1,360
CZ-8NT1	トラックボール	13,800	11,500
CZ-8PK10	24ドット136桁漢字プリンター	99,800	69,000
CZ-8PK7	24ドット80桁漢字プリンター	22,000	59,800
CZ-8PC4	24ドット熱転写カラー漢字プリンター	99,800	59,800
CZ-8TM1	モデムユニット300b	29,800	6,000
AN-8TU	RGBシステムチューナー	33,100	特価
AN-S100	アンプ付スピーカー	59,800	49,800
HXD040	40Mハードディスク(ITM)	118,000	95,000
HXD140	40Mハードディスク内蔵用(ITM)	98,000	79,800
MZ-14FD	カラーディスプレーアナログ0.31	49,800	特価
MZ-1D10	12"モノクロディスプレー	41,800	25,000
MZ-1D17	15"CRT mz-5500/6500/2	124,000	59,800
MZ-1E05	MZ-2000 FDインターフェイス	24,500	18,000
MZ-1E08	プリンターI/F 2000/2200/80B	9,000	8,000
MZ-1E11	MZ-6500用 SFD I/F	38,000	25,000
MZ-1E04	MZ-2000プリンター I/F	10,000	6,000
MZ-1E21	MZ-5500 GP I/F	36,000	12,000
MZ-1E18	MZ2000QD用インターフェイス	9,800	3,000
MZ-1E33	MZ6500パラレル I/F	34,800	28,000
MZ-1E45	MZ-6500 232C I/F	50,000	15,000
MZ-1E32	MZ2500 パラレルI/F	30,000	27,000
MZ-1E44	MZ-6500 S-RN I/F	50,000	15,000
MZ-1E22	MZ-5500 GPIB I/F	72,800	25,000
MZ-1E29	RS-232Cインターフェイス	17,800	9,800
MZ-1E01	MZ-3500 232Cボード	28,000	13,000
MZ-1E14	MZ1500QD用インターフェイス	9,800	3,000
MZ-1M01	MZ- 2000/220016ビットボード	78,000	8,000
MZ-1M09	MZ-6500 8082-2 演算プロセッサ	82,000	30,000
MZ-1M03	MZ-5500 数値演算	69,000	38,500
MZ-1M12	MZ-2861 8087 演算プロセッサ	90,000	45,000
MZ-1P06	ドットプリンター	234,000	45,000
MZ-1P28	ドットプリンター 漢字80桁	148,000	118,400
MZ-1P10A	24ドットプリンター 漢字80桁	245,000	79,000
MZ-1P22	熱転写漢字プリンター	59,800	25,000
MZ-1P29	漢字プリンター 136桁	168,000	134,400
MZ-1P30	136桁プリンター	228,000	120,000
MZ-1R01	MZ-2000/2200Gボード	39,800	10,000
MZ-1R10	MZ-5500 漢字ROM付	30,000	9,800
MZ-1R09	MZ-5500 V.RAM	35,000	15,000
MZ-1R06	MZ-5500増設RAM	45,000	8,000
MZ-1R12	MZ-80B/2000/1500/700 RAM	35,000	8,000
MZ-1R11	MZ-5500 256KRAM	80,000	35,000
MZ-1R36	MZ-2861IM 増設RAM	45,000	15,000
MZ-1R35	MZ-2861IM 増設RAM	55,000	19,000
MZ-1R14	MZ-5500 辞書ROM	40,000	22,000
MZ-1R16	MZ-5500 128KRAM	30,000	8,000
MZ-1R27A	MZ-2500VRAM	13,000	10,000
MZ-1R26	MZ-2500 増設RAM	15,000	12,800
MZ-1R21	漢字ROM	38,000	13,000
MZ-1R24	MZ-1500 辞書ROM	22,000	6,000
MZ-1R32	MZ6500RAM	80,000	40,000
MZ-1R31	漢字ROM	28,000	20,000
MZ-1R28A	MZ-2500 辞書ROM	13,000	10,000
MZ-1R29A	MZ-1P22増設RAM	32,000	12,000
MZ-1S13	MZ-1D17チルトスタンド	12,000	5,000
MZ-1T02	MZ-2200 データーレコーダー	19,800	8,500
MZ-1T03	MZ-5500 データーレコーダー	12,000	8,500
MZ-1U09	MZ-2500 拡張ボード	9,000	7,200
MZ-1V01	パソコン FAX	278,000	65,300
MZ-1X22	モデムユニット	21,800	13,000
MZ-2Z014	MZ-5500 TODAY	25,000	15,000
MZ-2Z016	MZ-5500 附属		5,000
MZ-2Z028	MZ-6500 MS DOS.GWBASIC	60,000	35,000
MZ-2Z023	MZ-5500 GWBASIC	50,000	30,000
MZ-2Z031	MZ-6500 日本語ワープロ	49,800	15,000
MZ-2Z029	MZ-6500 TODAY	68,000	20,000
MZ-2Z064	MZ-6500 書院RAM付	69,800	28,000
MZ-2Z065	MZ-6500 書院RAMなし	49,800	15,000
MZ-2Z012	MZ-5500 附属		5,000
MZ-2Z013	MZ-5500 MS DOS	25,000	20,000
MZ-4Z001	MZ-5500IBM変換ユーティリティ	30,000	8,000
MZ-5511	本体	288,000	35,000
MZ-5Z013	MZ-1500 QD通信ソフト		3,500
MZ-6BE2	X6800 2M RAM	35,000	23,500
MZ-6F03	ブランクQD DISK	450	400
MZ-6P18	MZ-IP18.28 カットシートフィーダー	60,000	35,000
MZ-6P11	MZ-IP10 カットシード	95,000	35,000
MZ-6P29	MZ-IP29 カットシートフィーダー	50,000	37,500
MZ-6P27	MZ-IP27 カットシートフィーダー	58,000	39,800
MZ-6P06	MZ-1P06 トラクターフィード	15,000	7,500
MZ-6P20	MZ-1P22/17ロールホルダー	3,100	2,700
MZ-6Z22	M-50 CP/M86	10,000	6,000
MZ-6Z25	M-50 ストリーマユーティリティZプロセッサ	39,800	15,000
MZ-80T20A	MZ-80 マシンランゲージ	6,000	5,000
MZ-80TUB	MZ-80 バックアップ	20,000	8,000
MZ-80P4B	136桁ドットプリンター		48,000
MZ-80TU	MZ-80 システムプログラム	20,000	8,000
MZ-80T40A	MZ-80 PASCAL	10,000	5,000
MZ-80T70A	MZ-80 FDOS	20,000	7,000
MZ-8BGK	MZ-80 BGRAM2	39,000	10,000
MZ-8BIO4	MZ200/2200. GP.IBインターフェイス	45,000	18,000
MZ-8BG	MZ-80 BGRAM1	39,000	10,000
MZ-8BC01	MZ200/2200. GP.IBケーブル	18,000	8,000
MZ-8BD02	MZ-80 BFDOS	50,000	18,000
PC-1280	ポケコン	24,800	19,600
PC-1248DB	ポケコン	11,000	9,800
PC-1262	ポケコン	24,800	19,600
PC-1360	ポケコン	29,800	19,800
PC-1360K	ポケコン	36,800	32,800
PC-1600K	ポケコン	69,800	49,800
PC-E500	ポケコン	28,800	19,800
PC-E550	ポケコン	32,000	特価

ポケコン関係周辺機器サプライ製品及シャープ関係のソフトウエア全種取扱います。

X68000 全機種取り揃え大特価セール

新店舗(京王線・北野駅前)オープン記念セール実施中！

SHARP X68000シリーズ対応 ハードディスク

〈ITEM〉

HXD 040 X68000
定価￥118,000⇨特価￥95,000

HXD 042 X68000 増設用
定価￥128,000⇨特価￥102,500

HXD 140 X68000 内蔵用
定価￥98,000⇨特価￥79,800

京王線・北野駅前店

ミニ電脳遊園地／12月8日(土)・9日(日)開催

マイコン、パソコン、ワープロ教室開催中！

SHARP X68000

※特価表示はTELにてご確認下さい。

CZ-603C(本体)

プラス(ディスプレイ)組合せ

機種	価格
CZ-602DBK	特　価
CZ-603D	特　価
CZ-611DGY	￥305,000
CZ-613D	特　価

CZ-652C(本体)

プラス(ディスプレイ)組合せ

機種	価格
CZ-602DBK	￥275,000
CZ-603DB	￥260,000
CZ-612DGY	￥290,000
CZ-605D	￥290,000

CZ-612CBK(本体)

プラス(ディスプレイ)組合せ

機種	価格
CZ-603DBK	￥330,000
CZ-605DBK	￥360,000
CZ-613DBK	￥370,000
CZ-602DBK	￥345,000

CZ-663C(本体)

プラス(ディスプレイ)組合せ

機種	価格
CZ-604D	特　価
CZ-611DGY	￥385,000
CZ-612DGY	￥400,000
CZ-605D	特　価

CZ-653C(本体)

プラス(ディスプレイ)組合せ

機種	価格
CZ-602DBK	特　価
CZ-603D	特　価
CZ-612DGY	￥290,000
CZ-605D	特　価

CZ-602C(本体)

プラス(ディスプレイ)組合せ

機種	価格
CZ-603DGY	￥270,000
CZ-613DGY	￥310,000
CZ-605DGY	￥300,000
CZ-611DGY	￥285,000

CZ-613CBK(本体)

プラス(ディスプレイ)組合せ

機種	価格
CZ-604D	￥410,000
CZ-605D	￥430,000
CZ-613D	￥440,000
CZ-21HD	￥450,000

CZ-623CTN(本体)

プラス(ディスプレイ)組合せ

機種	価格
CZ-611DGY	￥445,000
CZ-612DGY	￥460,000
CZ-613DTN	特　価
CZ-21HD	特　価

アイビット推奨ディスプレイ

●シャープCZ-860D・BK
カラーディスプレイ
0.31チルト付A/D 15/24
定価￥92,200⇨
特価￥59,800

CZ-860D対応パソコン機種：CZ880C/881C。X1/TURBOシリーズ。ケーブルは本体付属を使用。NEC PC-8801・9801シリーズ(XA・XLのみ不可) MZ700/1500/2000/2200/2500各シリーズ(推奨品シャープ8D8K)。(ドットヒッチ0.39)

●シャープ CZ-603D-GY・BK
(15型カラーディスプレイ)
ドットピッチ3.9
定価￥84,800⇨
特価

CZ-603D対応パソコン機種：※X1シリーズ/※X1 turboシリーズ/X1 turbo Zシリーズ/X68000シリーズ/PC8801シリーズ/PC-9801シリーズ/PC-286シリーズ
(※は接続ケーブルAN1506が必要です)

●シャープCZ-830D・BK
(14型)
2モードオートスキャン方式
(アナログ/デジタル)
定価￥98,000⇨
特価￥54,800《在庫限り》

CZ-830D対応パソコン機種：CZ880C/881C。X1/TURBOシリーズ。ケーブルは本体付属を使用。NEC PC-8801・9801シリーズ(XA・XLのみ不可) MZ700/1500/2000/2200/2500各シリーズ(推奨品シャープ8D8K)。

●シャープCZ-602D-BK
(15型アナログTV/3モードオートスキャン)
定価￥99,800⇨
特価￥75,000

CZ-602D対応パソコン機種：※X1シリーズ/※X1 turboシリーズ/X1 turbo Zシリーズ/X68000シリーズ/PC8801シリーズ/PC-9801シリーズ/PC-286シリーズ
(※は接続ケーブルAN1506が必要です)

●三菱XC-1498CII
(14型アナログ)
ドットピッチ0.28
定価￥107,000⇨
特価￥59,800

XC-1498CII対応パソコン機種：PC-9801シリーズ/PC-286シリーズ/PC-386シリーズ/PC-8801シリーズ
(上記機種には付属の接続ケーブルで、接続可能)

※シャープ周辺機器(拡張、プリンター他)も常時取り扱っております。

SHARP AX286L-F ラップトップ
定価￥428,000
➡在庫処分中！

SHARP AX286N-H2 All In Note
定価￥398,000
➡大特価！

NEC PC-9801n NOTE
定価￥248,000
➡特価￥198,000

TOSHIBA J3100SS Dyna Book
定価￥198,000
➡特価￥149,000

富士通FM TOWNSお買得セット

FM TOWNS モデル1基本セット

品名	価格
FM TOWNS-1	¥338,000
FMT-DP531	¥ 89,000
FMT-KB101	¥ 20,000
B276A010	¥ 20,000
特選ラック	¥ 24,000
定価合計	¥491,800

特価￥198,000

FM TOWNS モデル2基本セット

品名	価格
FM TOWNS-2	¥398,000
FMT-DP531	¥ 89,800
FMT-KB101	¥ 20,000
B276A010	¥ 20,000
特選ラック	¥ 24,000
定価合計	¥551,800

特価￥278,000

FM TOWNS モデル1拡張セット

品名	価格
FM TOWNS-1	¥338,000
HM-01T	¥ 32,800
FMT-DP531	¥ 89,800
FMT-KB101	¥ 20,000
B276A010	¥ 20,000
FMT-FD301	¥ 28,000
特選ラック	¥ 24,000
定価合計	¥552,600

特価￥258,000

FM TOWNS モデル1S基本セット

品名	価格
FM TOWNS-1S	¥338,000
FMT-DP531	¥ 89,800
FMT-KB101	¥ 20,000
B276A010	¥ 20,000
特選ラック	¥ 24,000
定価合計	¥491,800

特価￥238,000

FM TOWNS モデル1S拡張モデル2セット

品名	価格
FM TOWNS-1S	¥338,000
HM-01T	¥ 32,800
FMT-DP531	¥ 89,800
FMT-KB01	¥ 20,000
B276A010	¥ 20,000
特選ラック	¥ 24,000
定価合計	¥524,600

特価￥268,000

FM TOWNS モデル2Fセット

品名	価格
FM TOWNS-2F	¥378,000
FMT-KB101	¥ 20,000
FMT-DP531	¥ 89,800
B-276A010	¥ 20,000
特選ラック	¥ 24,000
定価合計	¥531,800

特価￥310,000

〈TOWNSお買い上げの方〉パソコン教室が御利用できます。初・中・上級者 無料にて実施中！

〈全商品新品完全保証付〉■シャープポケコン全商品販売中。カタログ、特価表ご請求ください(〒72)

☎0426-45-3002(駅前店)－3001(本 店)－3003(教 室)

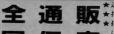
FAX.0426-44-6002

●営業時間/10:00~19:00●電話受付/20:00迄可●定休日/水曜日

SHARP SUPER XEX SHOP

アイビット電子株式会社 〒192 東京都八王子市北野町560-5

上記の広告商品はすべて店頭販売もしております

全国通販
北海道から沖縄まで

★送料はご注文の際にお問い合わせ下さい。
★掲載の商品は、すべて新品、保証書付きです。
★掲載の商品は充分用意してありますが、ご注文の際は、在庫の確認の上、現金書留または、銀行振込でお申し込み下さい。全商品クレジットでも扱っております。
★お申し込みの際は必ず電話番号を明記して下さい。
★商品、品切れの節はご容赦下さい。

富士銀行八王子支店 (普)1752505

●本誌発売時には上記価格よりさらにお求めやすい価格に変更されている場合があります。●この広告の商品にはすべて送料・消費税は含まれておりません。

全国通販

新品ソフト15%OFF 送料、消費税込み。

ただし、北海道・沖縄、離島の方は200円プラスして送金して下さい。
定価5,000円未満の商品についてはプラス300円。

PC98シリーズ

商品名	定価(円)	販売価格
三国志II	14,800	12,500
ポピュラス	9,800	8,300
MISTY6	5,000	4,200
ダンジョンマスター	9,800	8,300
シムシティー	9,800	8,300
大航海時代	9,800	8,300
続ダンジョンマスターカオスの逆襲	9,800	8,300
バーズテイル	9,800	8,300
キサナ	6,800	5,700
ドラゴンナイトII	6,800	5,700
機動戦士ガンダム デザート	9,800	8,300
サークII	8,800	7,400
D.P.S.SG	6,800	5,700
満開電飾	7,800	6,600
ヒルズファー	9,800	8,300
F15ストライクイーグルII	10,800	9,100
46億年物語	9,800	8,300
煩悩予備校	7,800	6,600
イルミナ	6,800	5,700
ダークレイスII	7,800	6,600
デジャップ1	8,800	7,400
ナビチューンドラゴン航海記	8,800	7,400
銀河英雄伝説II	9,800	8,300
ストロベリー大戦略	6,800	5,700
デ・ジャ	6,800	5,700
大戦略III'90	9,800	8,300
DUEL	8,700	7,300
アクティー	9,800	8,300
大戦略III「赤の逆襲編」	3,600	3,300
プリンスオブペルシャ	8,800	7,400
キャンペーン版大戦略IIマップ	4,800	4,000
麻雀悟空－天竺への道	9,800	8,300
クォータースタッフ	9,800	8,300
サイレントメビウス	14,800	12,500
BLACK RAINBOW	8,800	7,400
ごくらく天国おめみえの巻	9,800	8,300
ぷりんぐあっぷ	9,800	8,300
バトルチェス	9,800	8,300
D.P.S SG	6,800	5,700

3.5"版も在庫あります。

PC88シリーズ

商品名	定価(円)	販売価格
イルミナ	6,800	5,700
レイガン	6,800	5,700
キサナ	6,800	5,700
アズューライク	6,800	5,700
雀ボーグすずめ	7,800	6,600
大航海時代	9,800	8,300
天使たちの午後番外3	8,800	7,400
手天童子	8,800	7,400
ティルナノーグ	8,800	7,400
斬-夜叉円舞曲	9,800	8,300
サークII	8,800	7,400
ファンタジーIV	9,800	8,300
きゃんきゃんバニースペリオール	6,800	5,700
リップスティックADV II	6,800	5,700
DPS	5,400	4,500
トンネルズ&トロールズ	9,800	8,300
ランスII	6,800	5,700
ランペルール	9,800	8,300
ワールドゴルフIII	8,800	7,400

その他多数在庫あり

X68000シリーズ

商品名	定価(円)	販売価格
栄光は君に	9,500	8,000
天下統一	9,800	8,300
熱血高校ドッジボール部サッカー編	8,800	7,400
メネシス'90	8,800	7,400
ダンジョンマスター	9,800	8,300
続ダンジョンマスターカオスの逆襲	9,800	8,300
ポピュラス	9,800	8,300
ワンダラーズフロムイース	8,700	7,300
レインフォーサー	8,800	7,400
ジェミニウイング	8,800	7,400
ストロベリー大作戦	6,800	5,700
スーパーハングオン	8,800	7,400
三国志II	14,800	12,500
闇の血族（上巻）（下巻）	8,800	7,400
ワールドコート	8,800	7,400
幻獣鬼	8,800	7,400
シムシティー	9,800	8,300
クォース	6,800	5,700
ガンシップ	11,800	10,000
提督の決断	14,800	12,500
遥かなるオーガスタ	12,800	10,800
ラグーン	8,800	7,400
アンデッドライン	8,800	7,400
レイガン	6,800	5,700
機甲師団	9,500	8,000
AXIS	8,800	7,400
映画狂殺人事件	7,800	6,800

その他多数在庫あり

中古ソフト

中古リストご希望の方は62円切手3枚をお送り下さい。

フラント品	5.2HD	10枚	1,000円
ノーフラント	5.2HD	10枚	600円
ノーフラント	5.2D	10枚	400円
ノーブランド	3.5"2DD	10枚	600円
ノーフラント	3.5"2HD	10枚	1,500円

（消費税3%及び送料500円をプラスして送金して下さい。）

PC98シリーズ

商品名	定価(円)	販売価格
三国志II	14,800	
ポピュラス	9,800	
ダンジョンマスター	9,800	
サイレントメビウス	14,800	
キャンペーン版大戦略2	9,800	
トンネルズ&トロールズ	9,800	
FOXY	6,800	
ドラゴンナイト	6,800	
栄冠は君に	9,500	
インペリアルフォース	8,800	
ダークレイス	9,600	
エイトレイクスゴルフクラブ	4,800	
シムシティー	9,800	
フリンスオブペルシャ	8,800	
ドラゴンスレイヤーVI	8,700	
維新の嵐	9,800	
提督の決断	14,800	
水滸伝	9,800	
バトル	12,800	
ワンダラーズフロムイース	8,700	
46億年物語	9,800	
機甲師団	9,500	
戦略空軍	8,800	
ロンメル	8,800	
天と地と	12,800	
RYU	11,600	
ロードス島戦記	9,800	
プルトンレイ	8,800	
エメラルドドラゴン	9,800	
デジャ	6,800	
斬アナログ	9,800	
アークス2	9,800	

☎にてお問い合わせください。

3.5"版も在庫あります。

PC88シリーズ

商品名	定価(円)	販売価格
ドラゴンナイト	6,800	
FOXY	6,800	
DUEL	8,700	
ドラゴンスレイヤーVI	8,700	
信長戦国群雄伝	9,800	
水滸伝	9,800	
三国志II	14,800	
銀河英雄伝説	8,800	
大航海時代	9,800	
サバッシュ	7,800	
トンネルズ&トロールズ	9,800	
雀ボーグすずめ	7,800	
ソーサリアン	9,800	
イース1	7,800	
イース2	7,800	
イース3	8,700	
夢幻の心臓III	9,700	
きゃんきゃんバニースペリオール	6,800	
ストロベリー大戦略	6,800	
DPS	5,400	
維新の嵐	9,800	
アークス2	9,800	
ラストハルマゲドン	7,800	
ルーンワース	8,800	

その他多数在庫あり

☎にてお問い合わせください。

X68000シリーズ

商品名	定価(円)	販売価格
アースス2	9,800	
アールタイプ	7,800	
アフターバーナー	9,200	
イース3	8,700	
AXIS	8,800	
信長戦国群雄伝	9,800	
シムシティー	9,800	
グラナダ	8,800	
エージャックス	8,800	
ジェノサイド	8,800	
ナイトアームス	9,700	
サラマンダー	8,800	
スーパーハングオン	8,800	
天下統一	9,800	
ダンジョンマスター	9,800	
ポピュラス	9,800	
デスブリンガー	9,800	
大海令	12,800	
ラストハルマゲドン	9,800	
三国志II	14,800	
メタルサイト	8,800	
V'BALL	7,900	
源平闘魔伝	7,800	

その他多数在庫あり

☎にてお問い合わせください。

中古ソフト高価買取中!

- 代金は注文書を添えて、現金書留で送って下さい。（小為替不可）後払いシステムもあります。
- 新品ソフトをご注文の場合は、商品代金を送って下さい。（送料、消費税込み）
- 中古ソフトをご注文の場合は、必ず電話にて在庫確認をして下さい。
- 未発売ソフトの場合は、予約扱いとさせていただきます。
- 買取り希望の場合は、まずソフトを当店に送って下さい。こちらで高額査定のうえ、TELでご連絡させていただきます。値段が合わない場合、商品はすぐ返送しますので、安心してお送り下さい。
- ディスケットの送料は、100枚まで500円です。
- DISKシャトル フランチャイズ店募集開始。

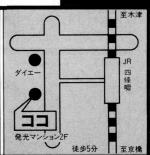
JR 高槻 北 明和ビル2F J&P 富士BANK 住友BANK 梅田 高槻 京都

DISKシャトル

営業時間 AM12:00〜PM8:00

高槻店 〒569 大阪府高槻市高槻町12－13 明和ビル2F
☎0726-83-9907

四條畷店 〒574 大阪府大東市北楠里27－19 発光マンション202
☎0720-78-1342

東京地区TELオープン☎03-713-1424

当社はX68000の販売認定店です。どんなことでも安心してご相談ください。

秋のX68000フェア実施中！ 即売・即納

X68000 NEW PRO II

- ●CZ-653C(本体)…………………¥285,000
- ●CZ-603D(カラーディスプレイ)………¥ 84,800
- ●お好きなゲームソフト1本…………¥ 7,800
- ■定価合計…………………………¥377,600

クリエイト特価

均等払い	¥ 7,680×48回	¥ 9,890×36回	¥14,370×24回
ボーナス	なし	なし	なし

X68000 EXPERT セット

5台限定！

- ●CZ-602C-GY(本体)………………¥356,000
- ●CZ-603D-GY(カラーディスプレイ)……¥ 84,800
- ■定価合計…………¥440,800▶大特価¥279,000

大特価¥279,000

均等払い	¥12,850×24回	¥ 8,870×36回	¥ 6,920×48回
ボーナス	なし	なし	なし

X68000 NEW EXPERT II

- ●CZ-603C(本体)………………¥338,000
- ●CZ-613D(カラーディスプレイテレビ)……¥135,000
- ●CZ-8NJ2…………………………¥ 23,800
- ●お好きなゲームソフト1本…………¥ 9,800
- ■定価合計…………………………¥506,600

クリエイト特価

均等払い	¥ 9,970×48回	¥12,840×36回	¥18,660×24回
ボーナス	なし	なし	なし

X68000 SUPER HD

- ●CZ-623C-TN(本体・キーボード・マウス)………¥498,000
- ●CZ-613D-TN(カラーディスプレイ)…………¥135,000
- ●CZ-6BP1……………………………¥ 79,800
- ■定価合計……………………………¥712,800

クリエイト特価

均等払い	¥ 7,320×48回	¥10,100×36回	¥13,450×24回
ボーナス	¥42,000× 8回	¥50,000× 6回	¥80,000× 4回

※本広告に掲載の全商品の価格について消費税は含まれておりません。

X68000 NEW EXPERT II

ミュージシャンセット。これもTMネットワークだよ〜！

- ●CZ-603C…………………………¥338,000
- ●CZ-605D…………………………¥115,000
- ●MU1.B(MIDIボード&ソフト)………¥ 39,800
- ●CM32L……………………………¥ 69,000
- ●グラナダ…………………………¥ 8,800
- ●JOYカード…………………………¥ 1,800
- ■定価合計…………¥572,400▶超特価¥458,000

X68000 NEW PRO II

ゲーマーズセット。遊んで暮らせるSET！

- ●PRO II CZ653C…………………¥285,000
- ●0.31CRT CZ603D………………¥ 84,800
- ●グラナダ…………………………¥ 8,800
- ●Y'S………………………………¥ 8,700
- ●ポピュラス…………………………¥ 9,800
- ●スーパーハングオン………………¥ 8,800
- ●エージャックス……………………¥ 8,800
- ●サーク……………………………¥ 8,800
- ●アールタイプ………………………¥ 7,800
- ●アナログJOYSTIC XE-1AP…………¥ 13,800
- ■定価合計…………¥445,100▶超特価¥353,000

X68000シリーズ用 周辺機器・ソフト オール超特価!!

型番	品名	定価	ソフト名	品名	定価
CZ-6VT1	カラーイメージユニット	¥ 69,800	MUSIC PRO	MIDI版	¥ 28,800
CZ-8NS1	カラーイメージスキャナ	¥188,000	MUSIC PRO-68K	マウスを使った楽譜ワープロ	¥ 18,800
CZ-6BE1A	IMB増設RAMボード	¥ 38,000	SOUND PRO-68K	サウンドエディタ	¥ 15,800
CZ-6BE2	2MB増設RAMボード	¥ 79,800	Sampling PRO-68K	AD PCMサンプリングエディタ	¥ 17,800
CZ-6BE4	4MB増設RAMボード	¥138,000	Musicstudio PRO-68K V.1.1	MIDIマルチレコーディングソフト	¥ 28,800
CZ-8NM3	マウス・トラックボール	¥ 9,800	OS-9/X68000	マルチタスクオペレーティングシステム	¥ 29,800
BF-68PRO	高性能CRTフィルター	¥ 19,800	PRO-68K	サイバーノート	¥ 19,800
CZ-6BP1	数値演算プロセッサ・ボード	¥ 79,800	PRO-68K	ステーショナリー	¥ 14,800
CZ-8NT1	トラックボール	¥ 13,800	Ccompiler PRO-68K	ソフト開発セット	¥ 39,800
CZ-6BM1	MIDIボード	¥ 26,800	Human 68K Ver2.0	開発ツールセット	¥ 9,800
CZ-8NJ2	アナログスティック	¥ 23,800	PIO-6BE1-A	内蔵1MRAM	¥ 25,000
CZ-6TU	パソコンチューナ	¥ 33,100	PIO-6BE2-2M	2MRAM	¥ 50,000
SX-68M	MIDI I/F	¥ 19,800	PIO-6BE4-4M	4MRAM	¥ 88,000
XE-1AP	アナログジョイパッド	¥ 13,800	MU1-B	MIDI I/F＋ソフト	¥ 39,800

▲上記以外ビジネスソフト、最新ゲームソフト豊富に在庫あります。※送料はご注文の際お問合せください。 ●超特価販売中！

★この表以外の組合せ、お支払い方法もご自由にできます。
★X1シリーズ用、X68000シリーズ用各社ハードディスク/プリンタ等の周辺機器を大特価にて販売しております。
電話にてお問合せください。

●横浜店
駅から歩いて2分！
相鉄ビル
三越
三和銀行
第一証券
ソフトクリエイト
横浜高島屋
三洋証券
岡田屋モアーズ
西口広場（ダイヤモンド地下街）
東京証券
東横線
横浜駅
横浜東急ホテル
京浜急行
国道1号線
至桜木町
至東京

オール15%〜20%OFF

総合お問合せ先☎03-486-6541(代)

パソコン専門ショップ

ソフトクリエイト 渋谷/横浜

●渋谷店☎03-486-6541(代)
〒150：東京都渋谷区渋谷1-12-7 三和渋谷ビル
振込銀行：三井銀行 渋谷宮益坂支店㊢No.5000340

●横浜店☎045-314-4777(代)
〒221：横浜市神奈川区鶴屋町2-12-8 第1建設ビル
振込銀行：三和銀行 横浜駅前支店㊢No.310852

SHARP

パソコンテレビ X1 turbo Z III

パーソナルコンピュータ+キーボード+マウス	CZ-888C-BK	標準価格	169,800円(税別)
14型カラーディスプレイテレビ	CZ-860D-BK	標準価格	92,200円(税別)
チルトスタンド	CZ-6ST1-B	標準価格	5,800円(税別)

クリエイティブマインドを刺激するAV機能 テレビ、ビデオ、ビデオディスクなどの映像を最大4,096色のリアルな画像で瞬時にグラフィック画面に取り込めるカラー画像デジタイズ機能を標準装備。4段階の量子化取り込み、42通りのモザイク取り込みなど多彩なトリック取り込み処理もサポート。さらにクロマキー合成、インターレーススーパーインポーズ、4,096色対応デジタルテロッパ機能、ステレオFM音源…先駆のAV機能がアートワークの領域をさらに拡げます。

AV指向の高水準ベーシックZ-BASIC搭載 多色グラフィック、カラー画像処理、ステレオFM音源、バンクメモリ対応など、ターボZシリーズが本来もつクリエイティブな機能をフルサポート。また豊富な画面モードで多色を駆使するときに便利なグラフィック用関数(HSV, RGB, HALF, CDOWN, CUP)も装備。さらにFM音源制御用ステートメントとしてX68000と命令コンパチの拡張MMLの採用によりスムーズな8音同時演奏を実現しています。

●メインメモリ128Kバイト標準装備、Z-BASICで最大576Kバイトまでサポート●1Mバイトの5インチフロッピーディスクドライブ2基搭載●JIS第1/第2水準準拠漢字、「システム・ユーザー辞書」を標準装備した高度な日本語処理機能●ニューデザインのマウス標準装備●X1ターボシリーズの豊富なソフト資産が活用できるコンパチブル設計●プリンタ、RS-232Cなど豊富なインターフェイスを装備●ドットピッチ0.39mmのハイコントラストブラウン管、15kHz/24kHzのデュアルスキャン方式採用14型カラーディスプレイテレビ(別売)

資料請求券 X1turbo ZIII Oh!X 12月

シャープ株式会社 ●お問い合わせは…シャープ㈱電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22号☎(06)621-1221(大代表)
電子機器事業本部液晶映像システム事業部第2商品企画部 〒162東京都新宿区市谷八幡町8番地☎(03)260-1161(大代表)

T4910217912569 雑誌 02179-12

Oh!X 12月号 1990年12月1日発行(毎月1回1日発行) 通巻104号 昭和58年11月2日第三種郵便物認可 ソフトバンク株式会社発行 Printed in Japan 定価580円(本体563円)